ADAPTIVE CONTROL OF THE MILLING PROCESS

By

Fariborz Talebzade Ordubadi

B.Sc. (Mechanical Engineering) S.U.T. ,Tehran

M.Sc. (Mechanical Engineering)University of California,Berkeley

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE STUDIES

MECHANICAL ENGINEERING

We accept this thesis as conforming

to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

March 1989

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Mechanical Engineering

The University of British Columbia
Vancouver, Canada

Date April 6, 89

DE-6 (2/88)

Abstract

Cutting forces in the milling process vary depending on the work-piece geometry and cutting parameters. When the cutting forces exceed a certain limit, the tool may break and cause damage to the work-piece and eventually to the machine tool. Adaptive cutting force control systems can be used to manipulate cutting operation parameters in order to keep the cutting forces at a safe level. Successful application of the method leads to increased metal removal rate and productivity in machining processes.

In this thesis, a second order transfer function is used to represent the time invariant dynamics of a research milling machine's feed drive servo system. The command feed velocity is the input and the actual feed is the output of the servo system. The actual feed manipulates the cutting forces which are modelled by a first order time varying dynamic system.

Three existing adaptive control methods have been designed to control the milling process. Adaptive Proportional Integral Derivative (PID), Pole-Placement and Model Reference Adaptive Control (MRAC) algorithms have been simulated and experimentally verified. It has been shown that when the dynamics of both the time invariant servo and the time variant cutting process are modelled correctly, the adaptive control algorithms can perform well. Simulations and experiments, which have been carried out with identical cutting conditions, show that PID and Pole-Placement controllers can be successfully applied to milling force control.

Table of Contents

List of Tables

List of Figures

Nomenclature

A: process denominator polynomial

A_m: model denominator polynomial

A_0: observer polynomial

a: axial depth of cut

a, b: digital filter parameters

a_1, b_0: estimated parameters of model

α_t: weighting factor

B_f: friction coefficient

B: process numerator polynomial

B_m: model numerator polynomial

C_1: model denominator polynomial

C_2: regulation dynamics polynomial

D: model numerator polynomial

d: time delay

δ: deflection of the tool

ε: plant-model error

$\bar{\varepsilon}$: auxiliary error

ε^f:filtered plant-model error

ε^*: augmented error

F: resultant force

F_{ax}: average force in longitudinal direction

F_{ay}: average force in transverse direction

F_r: radial force

F_t: tangential force

F_x: force in longitudinal direction

F_y: force in transverse direction

f: linear feeding velocity

G_{CL}: closed loop transfer function

G_{OL}: open loop transfer function

G_P: cutting process transfer function

G_{Pl}: plant transfer function

G_S: servo motor transfer function

H_g: tachometer constant

h: instantaneous chip thickness

J: equivalent inertia reflected to motor shaft

K: table's leadscrew gain

K_d: derivative gain

K_{da}: D/A converter gain

K_e: encoder gain

K_I: integral gain

K_P: proportional gain

K_p: digital filter gain

K_s: specific cutting pressure

K_t: torque constant of the motor

K_x: equivalent stiffness of the tool on the spindle

k: time index

L: gain vector

N: spindle speed

P: covariance matrix

ϕ: observation vector

ϕ_i: immersion angle

ϕ_S: swept angle

ϕ_T: cutter pitch angle

q^{-1}: backward shift operator

R, S, T: controller polynomials

R_g: current amplifier gain

r_1: ratio of radial to tangential forces

S_g: velocity command signal gain of the amplifier

s: Laplace transform variable

s_t: feed-rate

T: sampling period of control loop

T_1: sampling period of digital filter

θ: parameter vector

u: control input

u_c: reference input signal

u^M: reference input to model

ω_n: natural frequency of servo

y: process output

y^M: model output

Z: number of teeth on the cutter

z: Z transform variable

ζ: damping ratio

Acknowledgement

I would like to thank my supervisors Professors F.Sassani and Y.Altintas for their support in the development of this work. Professor Altintas stimulated my interest in this area and gave me valuable comments and suggestions for which I am grateful.

I would like to thank Research Engineers G. Rohling, P.K. Chan and G. Wright for helping to develop necessary software and hardware for this work.

I am indebted to my wife, Faati, whose support has made this work possible and also to my son, Kaveh, who sacrificed a lot of time so that this work could be completed.

This work was supported by NSERC of Canada under Grant Numbers OGP-0006164 and OGP-0005542.

Chapter 1

Introduction

The need for machining work-pieces of complex geometry with high precision led to development of numerically controlled (NC) and computer numerically controlled (CNC) machines. These machines relieved the operators from difficult and sometimes impossible tasks in machining parts. Nevertheless the problem of selecting cutting parameters, such as feed-rate and cutting speed has remained. The work-piece could be very complex. For example, consider the machining of an engine block with many holes and slots and variable axial depth and radial width of cut. Because of these variations the magnitude of the cutting force changes during the machining and if the peak force exceeds a certain limit, the tool breakage is almost certain. The problem is not limited to this and since the machine may still be moving, there is a great likelihood of damage to the work-piece and possibly the machine itself. The total loss regardless of labor cost for frequent tool changes and maintenance can be substantial. In order to save the tool part programmers usually take a conservative approach and choose cutting parameters according to the worst possible case, such as largest depth of cut, which exists in the work-piece in order to save the tool. It is clear that although the machine may be working at optimum conditions for that section, it will be slow for the rest of the job resulting in lower metal removal rate and productivity.

One way to overcome this problem is to design a controller in such a way that by manipulating the operating parameters one can maintain the desired specifications. Generally there are three approaches to design the controller: open loop adaptive

control, closed loop fixed gain and closed loop adaptive control. In open loop adaptive control the operating conditions are optimized in order to maximize a performance index. This kind of controller in manufacturing literature is referred to as Adaptive Control with Optimization (ACO). There are a few drawbacks in ACO systems such as difficulty in establishing a suitable performance index, and the need for better sensors and high computation power [1]. Fixed gain or conventional feed-back controllers are popular among manufacturing designers. The controller is a simple gain or at most a lead-lag compensator and the feed-back force is used to adjust the operating parameters to keep the cutting force at a constant level. This type of controller which operates well in certain circumstances will become unstable when cutting conditions such as axial depth of cut, material properties or tool condition change [23,2]. In closed loop adaptive systems it is assumed that the process parameters are unknown and time varying. Generally there are two control loops in this controller. The first one is a force feed-back loop as in the case of the conventional controllers. The other loop is designed to adjust controller parameters by getting information from input and output to the plant in such a way that the cutting force remains constant. The last two controllers are based on imposing a constraint, such as constant power or force, on the process. These controllers in manufacturing literature are referred to as Adaptive Control with Constraints (ACC).

In this thesis closed loop adaptive control is used to adjust feed-rate to keep the peak resultant force at a desired level. Therefore at a shallow depth of cut the milling table moves faster and vice versa. This way, the metal removal rate and productivity could be increased compared to the constant feed-rate operation[1,3,4,11]. On the other hand by keeping the cutting force at constant level, tool (shank) breakage could be reduced or eliminated. Peak resultant cutting force is the main cause for shank breakage but other factors such as torsional torque on the cutter, vibration and fatigue

can contribute to shank breakage. It is worth mentioning that by keeping the cutting force at a constant level, the tool deflection would remain constant both in low and high depths of cut, therefore the finished part would have better surface finish [40].

This thesis is organized as follows: in chapter two a brief history of NC and CNC machines is presented. Adaptive control is defined and different methods in the theory of adaptive control are discussed. Classification of adaptive control in manufacturing is discussed and finally a literature review of attempts to model and control the cutting process is presented. In chapter three cutting forces in milling are reviewed and a first order model based on static deflection of the tool is developed for the milling process. This model relates feed-rate to the peak resultant cutting force. To verify the derived model two experiments have been carried out and the results are presented. A description of experimental setup is also included in this chapter. The servomotor as part of the process is analyzed in chapter four and a second order transfer function is derived. A model of the cutting process is developed in chapter three which together with the model of the servo are used to design the controllers. Three different control algorithms, adaptive proportional-integral-derivative (PID)control, Pole-Placement Control and Model Reference Adaptive Control (MRAC) are designed , simulated and the results are shown. The algorithms are implemented on the milling machine and results are presented. A brief description of the Recursive Least Square (RLS) method for identifying the parameters of the process is included in chapter four. Finally in chapter five results of the thesis work are summarized and specific conclusions are drawn.

Chapter 2

Literature Review

2.1 Introduction

In this chapter, the history of Numerically controlled (NC) and Computer numerically Controlled (CNC) machines is briefly described and the application of adaptive control methods in metal removal processes is reviewed.

2.2 NC and CNC Machines

Development of NC and CNC machines was initiated after the World War II. The need for high precision parts in the U.S. Air Force promoted research for development of NC machines. It was clear that conventional machine tools could not satisfy the requirements of precision and complex parts. The first attempt was made by John Parsons of The Parsons Company in Michigan who presented methods of generating accurate smooth curves. In the late 1940's The Air Force asked the Massachussets Institute of Technology Servomechanism Laboratory to develop NC machines and eventually by 1952 the first NC milling machine with three controlled axes was constructed [1]. Development of NC machines, which was based on principles of digital electronics, was a revolution at that time in the machine tool industry. CNC systems, which replaced dedicated hard-wired control logic by a computer, were introduced in the early seventies. The trend away from NC to CNC systems means evolution from hardware-based to software-based equipment. Dedication of a computer to the system gives flexibility

in implementing more complicated control algorithms.

2.3 Adaptive Control

In this section development of adaptive control theory is briefly discussed and different schemes are explained. The idea of adaptive control(AC) was originated in the 1950s. A general definition for AC systems is : "Adaptive control systems are those which automatically adjust the controller settings to accommodate changes in the process to be controlled or its environment " [6]. As Astrom [5] describes, AC started with " a lot of enthusiasm, poor hardware, and non existent theory" and resulted in a flight disaster. Although the activities halted in this area for almost a decade, many contributions to control theory such as development of state space and stability theory, further improvement in stochastic control theory and system identification straightened the way. On the other hand the recent revolution in the electronics industry made it possible to implement adaptive control in a simple and inexpensive way.

In general there are three different schemes for adaptive control: gain scheduling, model reference adaptive control and self tuning control. All schemes consist of a control loop with an adjustable parameter controller, process and an ordinary feed-back loop. The main task is to change controller parameters in the presence of disturbances and / or changes in process dynamics to achieve control specifications. An excellent review of adaptive control is given in reference[7].

2.3.1 Gain Scheduling

Gain scheduling is used when an auxiliary variable can be found which correlates well with changes in the process dynamics(Fig.2.1). Then the parameters of the controller can be changed as a function of an auxiliary variable to accommodate changes in the

process gain. As this scheme was first used to set changes in process gain it is called gain scheduling. To apply the gain scheduling scheme a thorough knowledge of the physics of the system in question is required to find the auxiliary variable. Another drawback of gain scheduling is that the adaptation part of the control is open loop therefore the system is like a feed-back control where controller parameters are set by feed-forward compensation.

2.3.2 Model Reference Adaptive Control (MRAC)

Model reference adaptive control was originated by Whitaker, Yarmon and Kezer in 1958 [8]. The idea is to find a control input in such a way that the output of an unknown plant asymptotically approaches that of a given reference model (Fig.2.2). The difference between the output of the process and that of the model is the error signal. This error signal along with the plants inputs and outputs are used by an adjustment mechanism to change controller parameters and reduce the error signal to zero. There are two approaches in MRAC: direct and indirect. In the direct method the controller parameters are calculated directly from the adjustment mechanism. In the indirect method first the values of process parameters are estimated and then the controller parameters are calculated based on those values. The second approach has the advantage of availability of the process parameters.

2.3.3 Self-Tuning Control (STC)

A schematic diagram of self tuning control is shown in Figure2.3. The system consists of two loops: The inner loop is a simple feed-back loop and the outer loop is used to update the controller at each sampling time. Here the task is to estimate the parameters of the process recursively with an assumed dynamic model from input and output data. The estimated parameters and process model are then assumed to be exactly correct.

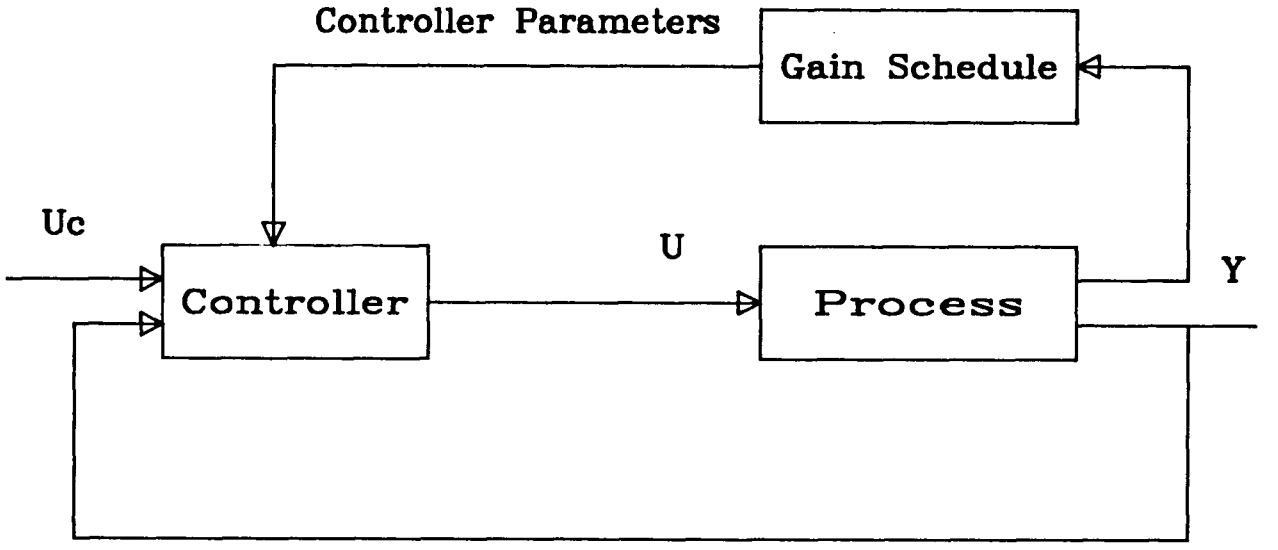

Figure 2.1: Block Diagram of Gain Scheduling Method

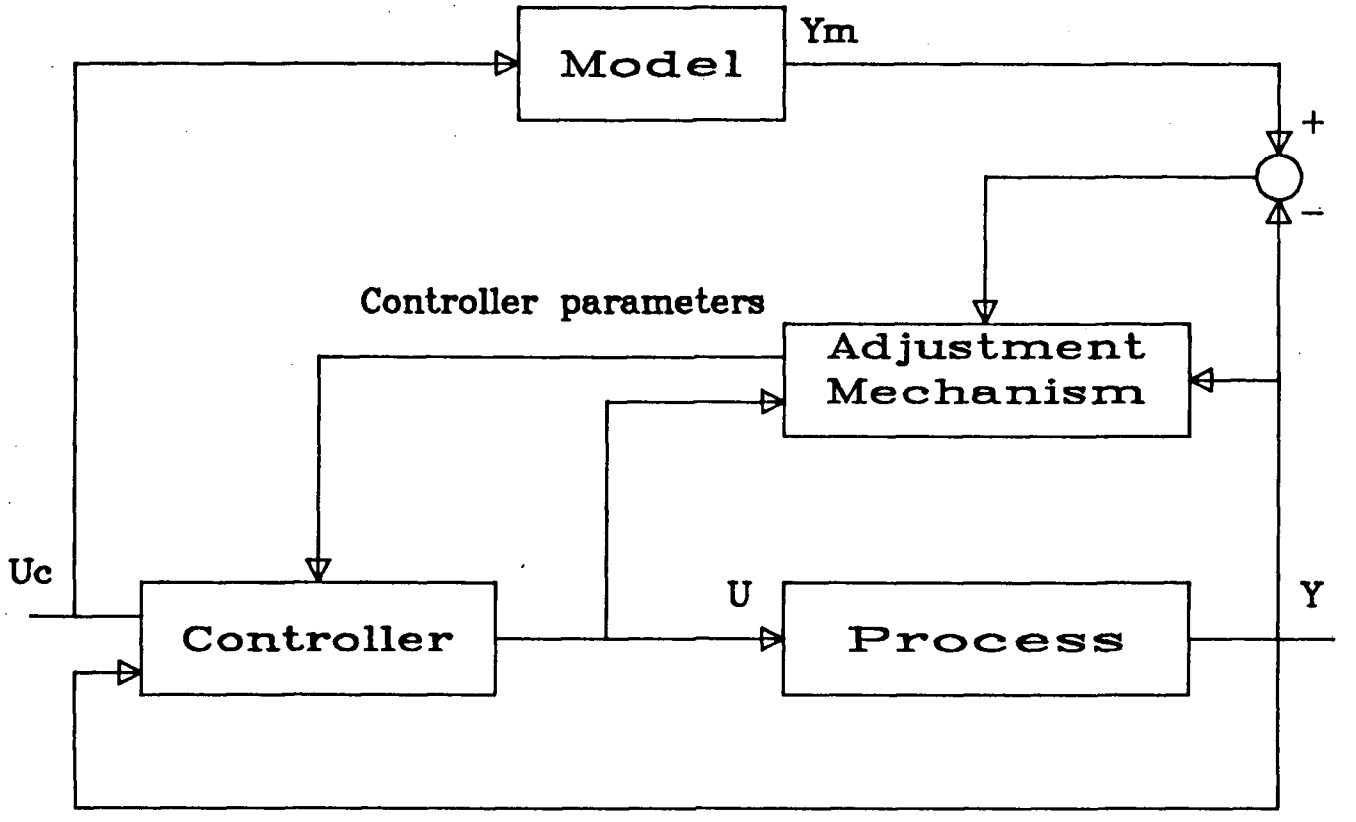

Figure 2.2: Block Diagram of Model Reference Adaptive Control Method

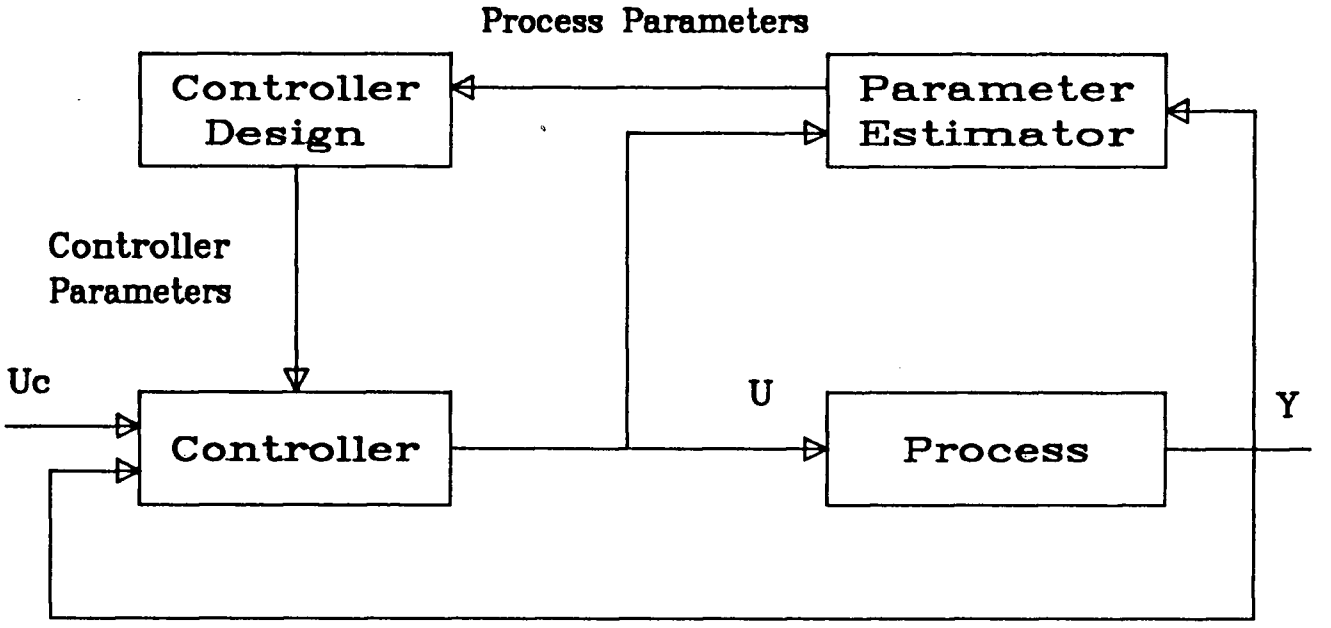

Figure 2.3: Block Diagram of Self-Tuning Control Method

This is called "certainty equivalence principle" and controller parameters are changed by the control algorithm. STC was first developed by Kalman in 1958 [9] and due to application of different estimators and control algorithms there are variations of this scheme in literature.

2.4 Adaptive Control of Metal Cutting Process

Application of adaptive control schemes in industrial processes is a fairly recent development. Before reviewing the literature it is necessary to clarify the meaning of adaptive control from two different view points. In the past, manufacturing literature considered the simple feed-back control as an adaptive scheme. It is clear that this approach was completely different from what was stated in the previous section. Feedback controllers are mainly simple gains and they will become unstable in response to changes in process dynamics. In this thesis adaptive control is analyzed from a control engineering point of view. Adaptive control systems for machine tools can also be classified into three categories : Geometric Adaptive Control(GAC), Adaptive Control with Optimization (ACO) and Adaptive Control with Constraints (ACC).

In GAC the machining parameters such as feed-rate or spindle speed are changed in such a way that the required geometry or accuracy of the work-piece is maintained. There are two general categories in GAC. In the first, there are controllers which change machine parameters such that the tool deflection remains at a certain level . In the second category, inaccuracies in the machine tool such as spindle or guideway errors are measured before hand or on line and then the tool path is compensated according to this information.

The idea in Adaptive Control with Optimization is to find a performance index,

usually in the form of a cost function, which is then optimized subject to the machining constraints. For example, in milling, sensors measure cutting conditions such as tool temperature, feed, torque, etc. Then these values are used to calculate power consumption, tool wear or metal removal rates. Next a performance algorithm is set up based on calculated values to find the performance index. Using the optimization algorithm and the process constraints the performance index is then maximized to yield operating conditions such as feed-rate or spindle speed. Constraints could be chosen as maximum torque, feed, temperature, maximum and minimum spindle speed etc.

Research in the area of ACO started with Centner and Idelson[10] in the early 1960's. They tried to model tool wear rate as a function of tool edge temperature and the rate of change of cutting torque. The work revealed the problems associated with ACO systems as follows: 1) more knowledge of the cutting process is needed to develop an accurate model of tool wear. This model definitely should relate more than two variables as they used. 2) better and more accurate sensors are needed to collect data. For instance, at that time, they could measure the temperature at tool work-piece contact area rather than the flank side. 3) ACO algorithms are more sophisticated than ACC algorithms, therefore more computational power is required.

In another case Bendini and Pinotti[11] tried to apply ACO system on a peripheral milling process. Their cost function was based on width and depth of cut, spindle speed, feed-rate and tool wear rate. The constraints they chose were bending moment on the tool, spindle torque and cutting power. They cited difficulties in measuring bending moment on the tool.

Because of aforementioned problems ACO has not been widely accepted in industry (except in grinding process[14]). Nevertheless research activities in this field have continued. Lately, Koren[12] developed a new model for tool wear and Yen and Wright[13] tried to combine ACC and ACO approaches in a unified manner.

Recently Watanabe[15] applied ACO system on a model of the end milling process. He identified shear stress and work-piece hardness, shear angle and true contact area at the flank wear land along with axial and radial depth of cut. Then he calculated the tool wear rate based on identified parameters. He claimed that ACO system gives higher cost efficiency than ACC when process parameters change.

In this thesis application of ACC systems on milling has been studied. In ACC systems the machining parameters are maximized subject to process and system constraints. For example in this work feed-rate is manipulated through the control algorithms such that the resultant cutting force remains at a level below the tool breakage limit. ACC systems have simpler algorithms compared with ACO systems and utilize low computational power. On the other hand sensors needed for ACC are fewer and relatively simpler.

One drawback of ACC systems is that the number of constraints used is limited and other constraints are treated indirectly. For example in this work the constraint is cutting force to prevent shank breakage. But other important aspects such as tool breakage, tool wear, surface finish, etc. are not addressed here. These are important issues which could be considered in a further extension of this work.

The idea of using ACC systems with different control strategies in metal cutting has been examined by several researchers. Beadle and Bollinger[16] used an ACC system for a face milling machine. They considered the whole system as a gain with one sampling time delay. Then, they designed the controller with a dead-beat closed loop response. Finally to find the gain of the system they imposed a constraint on maximum power consumption of the machine and manipulated both spindle speed and feed-rate to keep the critically damped system at the stability boundary.

Stute and Goetz[17] applied ACC system for turning and milling process. They used a closed loop proportional-integral controller with the control input sent to the

process and a model which was parallel to the process. For example in turning they used a fourth order model comprising of a first order servo feed drive, second order cutting process and a first order main drive. They were concerned about rapid changes in process gain because of changes in axial depth of cut. They compared the output from the model and the process and by simple division found the gain of the controller. They used the same technique for milling but included a smoothing filter for undulations inherent in the milling process.

In another approach Tlusty and Elbestawi[18] applied ACC on an end milling process. They developed the transfer function of the servo as a third order system and used a simple gain with one tooth period delay as the cutting process dynamics. They simulated this system with a constant gain and also a PI controller such that the force could be kept constant by manipulating the controller gain.

Later Ulsoy, Koren and Rasmussen [14]summarized the previous works in the area of ACC systems and for the first time they distinguished between two different views of control as discussed before. They presented simulations for a first order turning process and a second order servo loop dynamics. For the controller they used an integral controller and by using a parallel model they manipulated the controller gain to keep the actual gain of the system constant(same as [17]). This algorithm which is simple, was applied later by Masory and Koren[20] on a lathe. They had to find the critical gain of the system at the limit of stability before hand to ensure stability. On the other hand selecting a lower gain to guarantee the stability is harmful too because it would result in a sluggish transient response which might damage the tool.

Regarding the stability of the whole process it is worth mentioning that up to now the changes in cutting process dynamics (the parameters of the process) are either neglected by assuming the process as a simple gain or is not identified from the control engineering point of view. In a manufacturing environment identification means

nothing but simple algebraic calculations based on empirical formulations. But process identification for system dynamics defined by a discrete auto-regressive moving average model is to find all the parameters of the model at every sampling time while the process dynamics are changing. By identifying the process parameters, the control theory could be used to ensure the overall stability of the system in a correct way.

The first comprehensive approach toward application of adaptive control theory on machine tools was by Tomizuka et al.,[19]. They developed a first order discrete model of milling process and since their machine feed drive was a stepping motor they considered it as a gain. They applied model reference adaptive control design based on Landau and Lozano's [21] scheme. There were two unknown parameters in the model of the cutting process. The first one was the gain of the process and the second one was related to the time constant. They noticed that the second parameter is almost constant. Therefore they fixed this parameter and tried a single parameter adaptation algorithm. This indicated that a good estimate of the gain of the system is essential and proved the work of previous researches [17,20,18]. One drawback of their work is that their reference force was about 50 Newton which is below the friction forces during many cutting operations.

Later Daneshmand and Pak[22] applied MRAC scheme[21] on a turning machine. They used a first order dynamics for the cutting process and servo together. They also used a nonadaptive feed-back loop with a PI controller around the system, ie. instead of reference force, the signal from the PI controller was sent to the model. Then through a series of tests by changing the regulation dynamics parameters and covariance matrix elements (see chapter four) they found the optimum operating condition. Their best result was also with single parameter adaptation. They filtered the force signal with a 5 Hz low pass filter but did not use its dynamics in the control loop.

At the same time Laderbaugh and Ulsoy[23] applied MRAC on a milling machine.

Their scheme was based on the identification of the model parameters and application of model following controller. They developed a non-linear second order differential equation for the process based on experiments and neglected the servo dynamics as did Tomizuka et al.,[19]. They noticed that run-out has a detoriating effect on the controller and therefore passed the force signal through a low pass filter.

Elbestawi and Sagherian[24] simulated several AC algorithms for an end mill process. They considered the servo as a third order dynamics with fixed parameters based on previous research[18] on their machine. They also used a first order linear (and in one case non-linear) model for the cutting process. Their simulations showed that with linear model of the cutting force dead-beat controllers of an increased order and discrete PID controller have superior performance. For the non-linear model they used a dead-beat controller. Recently Mohamad et al.[25] applied the latter controller under the same conditions to a milling machine. They showed that the controller was capable of handling changes in process dynamics. For the first time they introduced a stochastic part in their model to account for transducer and environmental noise along with noise due to cutter run-out.

Another approach to apply MRAC system was done by Fussel and Srinivasan [26]. They considered the end milling process as a gain and with one tooth period delay and used a parameter identification method to estimate the only parameter in the process. The servo was treated as a second order dynamic system and coupled with cutting dynamics represented the model. They used a series-parallel reference model approach based on [27]. They noticed some problems associated with run-out and indicated that have used an algebraic scheme to identify run-out on-line and then subtract the measured force from the run-out component of the force. In their work the force signal was filtered with a low pass second order filter before being sent to the adaptive controller. They used a non-adaptive feed-back loop similar to Daneshmand

and Pak[22] around the AC control loop with and without a PI controller. The reason to use PI controller is to minimize the error signal from the reference and measured signals. They obtained better results from the PI controller with fixed parameters.

Chapter 3

Milling Process

3.1 Introduction

Cutting forces generated in milling are modelled in this chapter. The transfer function of the milling process which is used for the development of control algorithms in chapter four is derived. Finally the experimental set-up is described and an experimental verification of the model is presented.

Milling is a cutting process in which the work-piece is sheared by the cutter and discontinuous chips are formed. Cutter is single-tooth or multi-tooth where the cutting edges are usually equally spaced around the cutter. Cutting is performed by rotation of cutter by means of the spindle while the work-piece, which is clamped onto the table, moves. Milling is a complex process because of interruptions in cutting which results in periodical variations of cutting forces.

3.2 Modelling of Cutting Forces in Milling

The resultant cutting force in milling has three components: tangential, radial and axial. The axial component is negligible. Figure 3.4 shows forces acting on a work-piece. The same forces but in opposite direction act on the tool. The tangential force is proportional to the axial depth of cut and the chip thickness.

$$F_t = K_s \, a \, h \tag{3.1}$$

where

K_s	specific cutting pressure
a	axial depth of cut
h	instantaneous chip thickness

The radial force is proportional to the tangential force.

$$F_r = r_1 F_t \tag{3.2}$$

Where r_1 is the radial to tangential force ratio. The resultant force is:

$$F = \sqrt{F_t^2 + F_r^2} \tag{3.3}$$

Specific cutting pressure which depends on tool-material geometry is a function of chip thickness

$$K_s = \acute{K}_s \, h^{-c} \tag{3.4}$$

where $\acute{K}_s$ and c are tool- material constants. Equation 3.4 shows the nonlinear nature of the cutting forces. This nonlinearity which is significant at low feed-rates is due to the flank and nose forces on the tool.

Chip thickness variation in milling is shown in Figure 3.4. The feed-rate is defined as the linear travel of one tooth in one revolution of the cutter.

$$s_t = \frac{f}{N\,Z} \tag{3.5}$$

where f is the linear feeding velocity(mm/min) , N is the spindle speed(rev/min) and Z is the number of teeth. Note that the tooth path is considered as a circular arc which is close to the actual torchoidal path. Instantaneous chip thickness is related to feed-rate as

$$h(\phi) = s_t \, \sin(\phi) \tag{3.6}$$

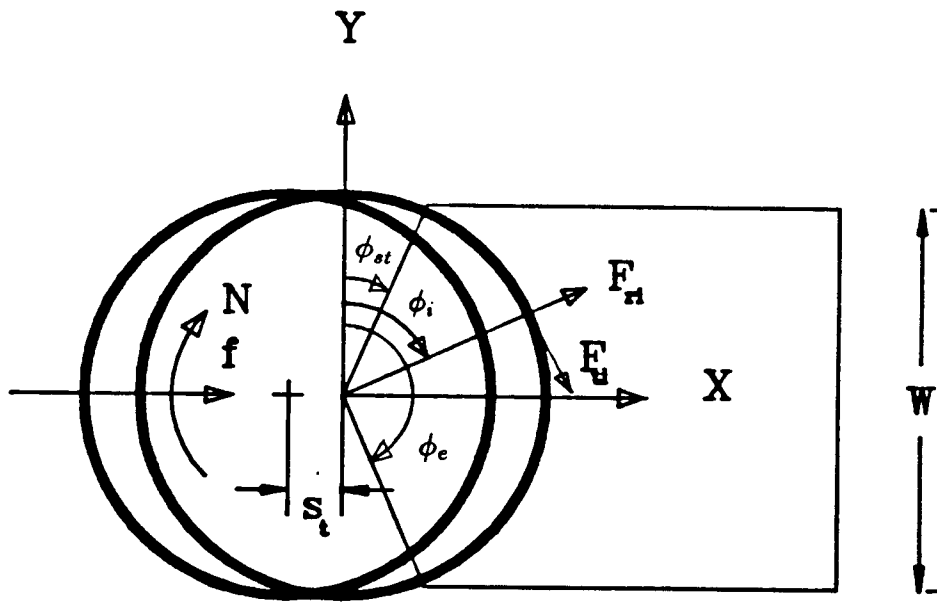

Figure 3.4: Chip Thickness Variation in Milling

where ϕ is immersion angle. Substituting Eq.3.6 into Eqs.3.1 and 3.2 yields

$$F_t = K_s \, a \, s_t \, \sin(\phi) \tag{3.7}$$

$$F_r = K_s \, a \, s_t \, r_1 \, \sin(\phi) \tag{3.8}$$

The components of the cutting force in cartesian coordinates are expressed as

$$F_x = F_t \, \cos(\phi) + F_r \, \sin(\phi) \tag{3.9}$$

$$F_y = F_t \, \sin(\phi) - F_r \, \cos(\phi) \tag{3.10}$$

Resultant cutting force from a simulation of two teeth and four teeth cutters are shown in figure3.5. Both simulations were performed for half immersion up-milling under identical cutting conditions. As it is shown the resultant cutting force is periodic at tooth passing frequency. There are identical wave forms as the number of teeth in cutter in every revolution. Peak forces are important and must be constrained to avoid tool breakage.

3.3 Modelling of the Milling Process

In this section a transfer function model for milling process is presented which is further extension of early works by Tlusty [28] and Tomizuka et al. [19]. This model which is an essential part for designing controller and simulation studies, is a difference equation which relates the response of cutting forces to variations of feed-rate for a two teeth cutter. A number of assumptions are necessary before developing the model:

- The model is derived for full slotting condition.
- Cutting process is sampled at tooth periods.
- Tool is flexible and deflects mainly in feed direction.

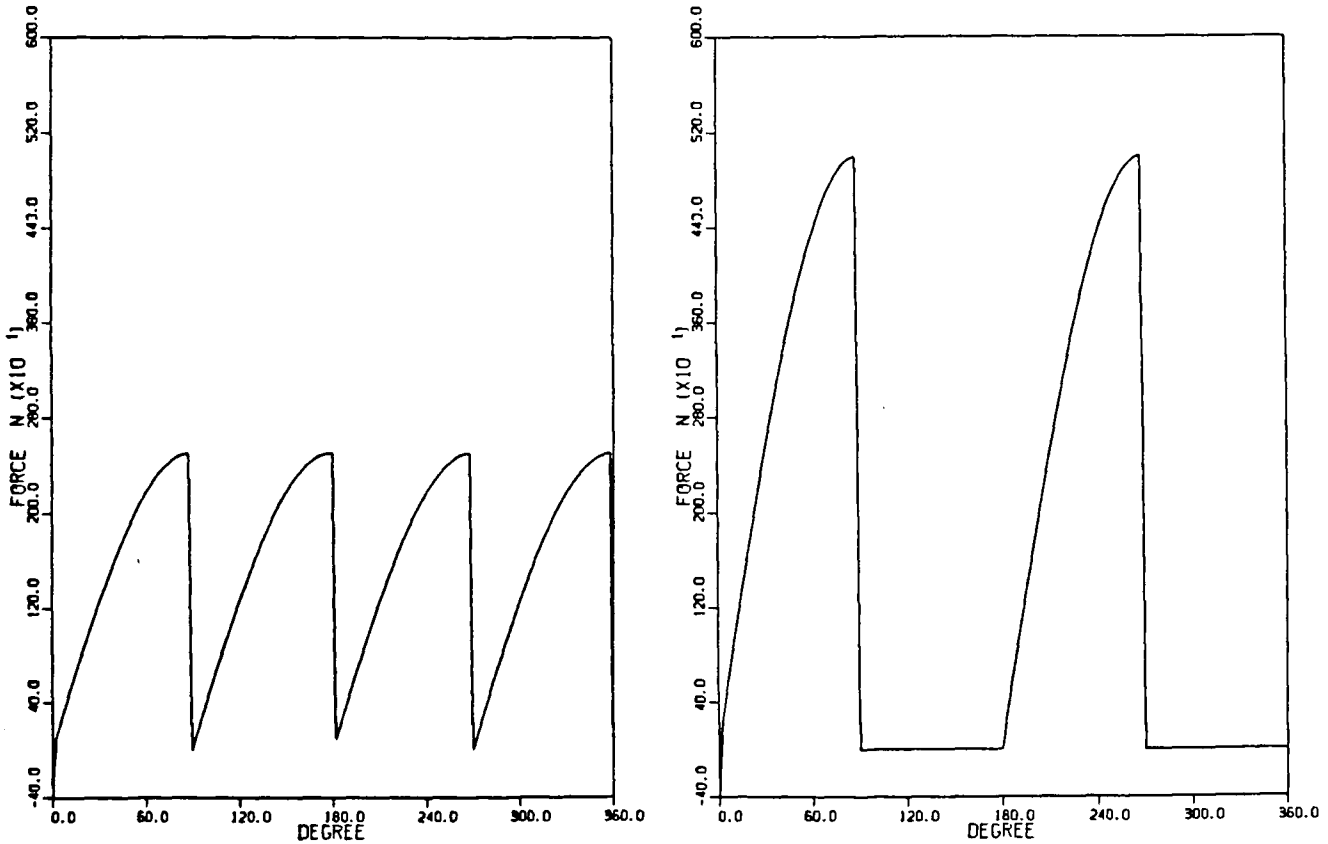

Figure 3.5: Resultant Force Simulation in Half Immersion Cutting
a) Two Teeth Cutter
b) Four Teeth Cutter
Entry Angle Zero, Exit Angle 90 Degrees, K_s = 2000 N/mm^2, ADC=25 mm, Feed-rate 0.2 mm/rev

- Edge forces are neglected assuming the tool is sharp.

In Fig. 3.6 commanded position of the tool is shown with dashed line while the solid line is the actual position of the tool after deflection. The tool and work-piece are shown at 180 degrees immersion angle windows. Actual chip thickness at tooth period$(i+1)$ is derived from the geometry (Fig.3.6) as,

$$h_{i+1} = s_{ti} + \delta_i - \delta_{i+1} \tag{3.11}$$

where

h_{i+1}	resultant chip thickness
s_{ti}	feed-rate/rev/tooth
δ_i	deflection due to F_{ri}

F_{ri} and F_{ti} are the maximum radial and tangential forces at each instant

$$F_{ti+1} = K_s\, a_i\, h_{i+1} = K_s\, a_i\, (s_{ti} + \delta_i - \delta_{i+1}) = F_{yi+1} \tag{3.12}$$

$$F_{ri+1} = r_1\, K_s\, a_i\, h_{i+1} = r_1\, K_s\, a_i\, (s_{ti} + \delta_i - \delta_{i+1}) = F_{xi+1} \tag{3.13}$$

The radial force deflects the end mill in the opposite direction of the feeding velocity

$$F_{ri} = K_x\, \delta_i \tag{3.14}$$

where K_x is the equivalent stiffness of the tool on the spindle. Equating equations 3.13 and 3.14

$$K_x\, \delta_{i+1} = r_1\, K_s\, a_i\, (s_{ti} + \delta_i - \delta_{i+1}) \tag{3.15}$$

Rearrangement of the above equation leads to

$$\delta_{i+1} = \frac{r_1\, \mu}{1 + r_1\, \mu}\, (s_{ti} + \delta_i) \tag{3.16}$$

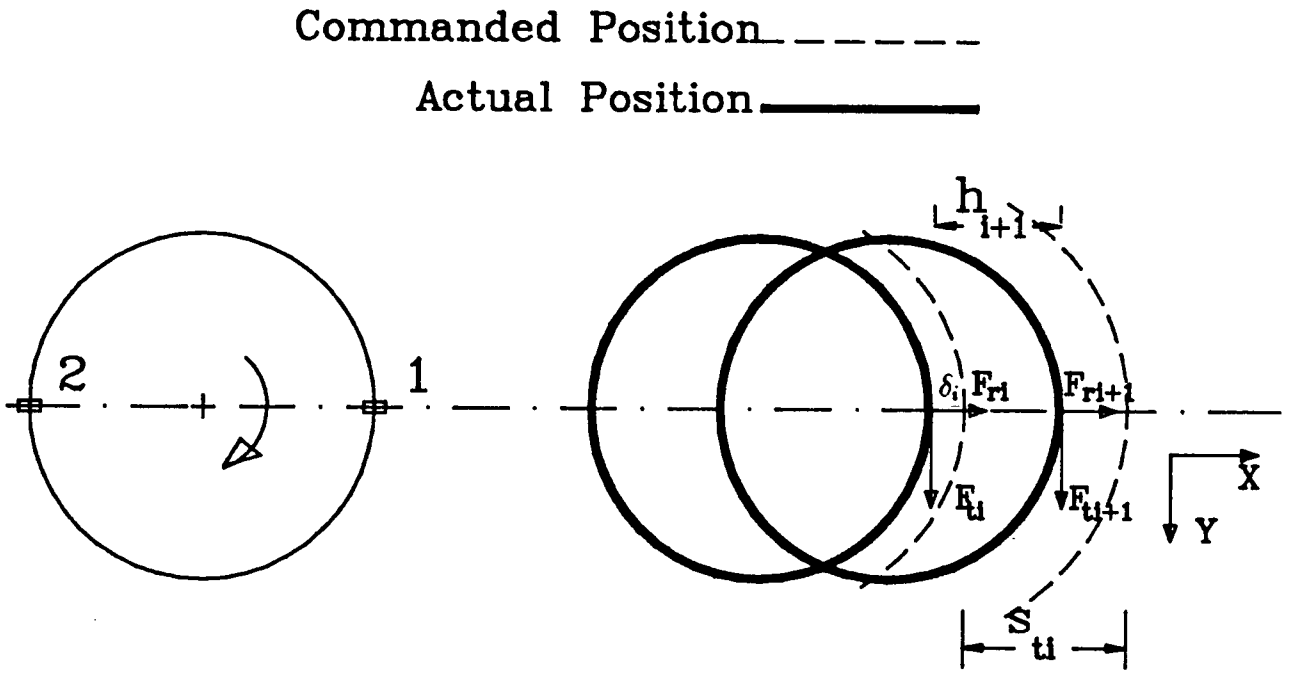

Figure 3.6: Modelling of Cutting Process in Milling

where

$$\mu = \frac{K_s\, a_i}{K_x} \tag{3.17}$$

substituting equations 3.14 into 3.16

$$F_{ri+1} = \frac{K_x\, r_1\, \mu}{1 + r_1\, \mu}(s_{ti}) \; + \; \frac{r_1\, \mu}{1 + r_1\, \mu}\,(F_{ri}) \tag{3.18}$$

F_{ri} and F_{ri+1} are present and future values of the radial force and are related by the forward shift operator (z) as

$$F_{ri+1} = \; z\; F_{ri} \tag{3.19}$$

Substituting equation 3.19 into 3.18 gives

$$F_{ri}\,(z \; - \; \frac{r_1\, \mu}{1 + r_1}) \; = \frac{K_x\, r_1\, \mu}{1 + r_1\, \mu}\,(s_{ti}) \tag{3.20}$$

Maximum resultant force (F_i)is derived from equations 3.3 and 3.19

$$F_i = \sqrt{F_{ti}^2 \; + \; F_{ri}^2} \; = \; \sqrt{\frac{F_{ri}^2}{r_1^2} \; + \; F_{ri}^2} \; = \; \frac{F_{ri}}{r_1}\,\sqrt{1 + r_1^2} \tag{3.21}$$

substituting equation 3.21 into 3.20 one gets

$$\frac{F_i}{s_{ti}} \; = \; \frac{[K_x\, \mu/(1 + r_1\, \mu)]\,\sqrt{1 + r_1^2}}{z \; - [\, r_1\, \mu/(1 + \; r_1\, \mu)]} \tag{3.22}$$

Equation 3.22 is the dynamic model of feed-force in milling process which is a first order system. The pole of the process is $\frac{r_1\, \mu}{1 + r_1\, \mu}$ which is always less than one. It is clear that if $r_1\, \mu \; \ll 1$ or $r_1\, K_s\, a_i/K_x$ is very small then the milling process could be considered as a simple gain with one step delay. This is the situation when the axial depth of cut is small or the tool is very rigid.

The transfer function derived above is for a special case where only one tooth is cutting at a time. However it can be extended to situations where more than one tooth

are in cut. In general for rigid milling the maximum cutting force F_i can be expressed as

$$F_i = K_s\, C_1(\phi_i\, , Z)\, \sqrt{1 + r_1^2}\, a_i\, s_{ti} \tag{3.23}$$

where $C_1(\phi_i\, , Z)$ is a parameter which depends on the current immersion (ϕ_i) and the number of teeth in cut. A time varying cutting constant which is common for any immersion and number of teeth can be expressed as

$$K_{ci} = K_s\, \sqrt{1 + r_1^2}\, C_1(\phi_i\, , Z) \tag{3.24}$$

The cutting constantK_{ci} changes as a function of immersion (ϕ_i) which may vary with time. The maximum force (F_i) is related to the tool deflection as

$$F_i = K_i\, \delta_i \tag{3.25}$$

where K_i is oriented stiffness in the direction of the feeding velocity. Note that since the immersion may change with time as a function of the machined work-piece geometry, the oriented stiffness K_i may also be time varying. Equating two cutting forces yields

$$K_{ci}\, a_i\, s_{ti} = K_i\, \delta_i \tag{3.26}$$

It is assumed that the oriented stiffness K_i changes gradually (ie. gradual immersion change)

$$K_i \simeq K_{i+1} = K \tag{3.27}$$

s_{ti} is replaced by the actual equivalent feed-rate of the cutter centre from Eq. 3.11

$$K\, \delta_{i+1} = K_{ci}\, a_i(s_{ti} + \delta_i - \delta_{i+1}) \tag{3.28}$$

or

$$\delta_{i+1} = \frac{\mu}{1 + \mu}\, (s_{ti} + \delta_i) \tag{3.29}$$

where

$$\mu = \frac{K_{ci}\, a_i}{K} \tag{3.30}$$

The maximum cutting force (F_{i+1}) due to deflection δ_{i+1} at tooth period $(i+1)$ can be expressed as

$$F_{i+1} = K\,\delta_{i+1} = \frac{K\,\mu}{1+\mu}\,(s_{ti} + \delta_i) = \frac{K\,\mu}{1+\mu}\,s_{ti} + \frac{\mu}{1+\mu}F_i \tag{3.31}$$

The discrete transfer function of the milling process due to static deflection of the tool is derived from Eq.3.31 as

$$\frac{F(z)}{s_t(z)} = \frac{b_0}{z + a_1} \tag{3.32}$$

where

$$\begin{aligned} b_0 &= \frac{K\,\mu}{1+\mu} \\ a_1 &= -\frac{\mu}{1+\mu} \end{aligned}$$

Note that the process parameters b_0 and a_1 are time varying and a function of the work-piece geometry. For example, during the machining of engine blocks or aircraft wings, both width of cut (i.e. immersion) and depth of cut may change according to work-piece and selected tool path geometry.

3.4 Experimental Set-up

A Superior Electric Slo-Syn Knee type vertical milling machine was used for the cutting test(Figure3.7). The machine was retrofitted with three BALDOR pulse-width-modulated (PWM) permanent magnet DC motors [31]. Each motor has tachometer and encoder devices for velocity and position feed-back respectively(Figure3.8). The encoder has a resolution of 1000 counts per revolution in each of two channels in quadrature which gives a combined resolution of 4000 counts per revolution.One count is the

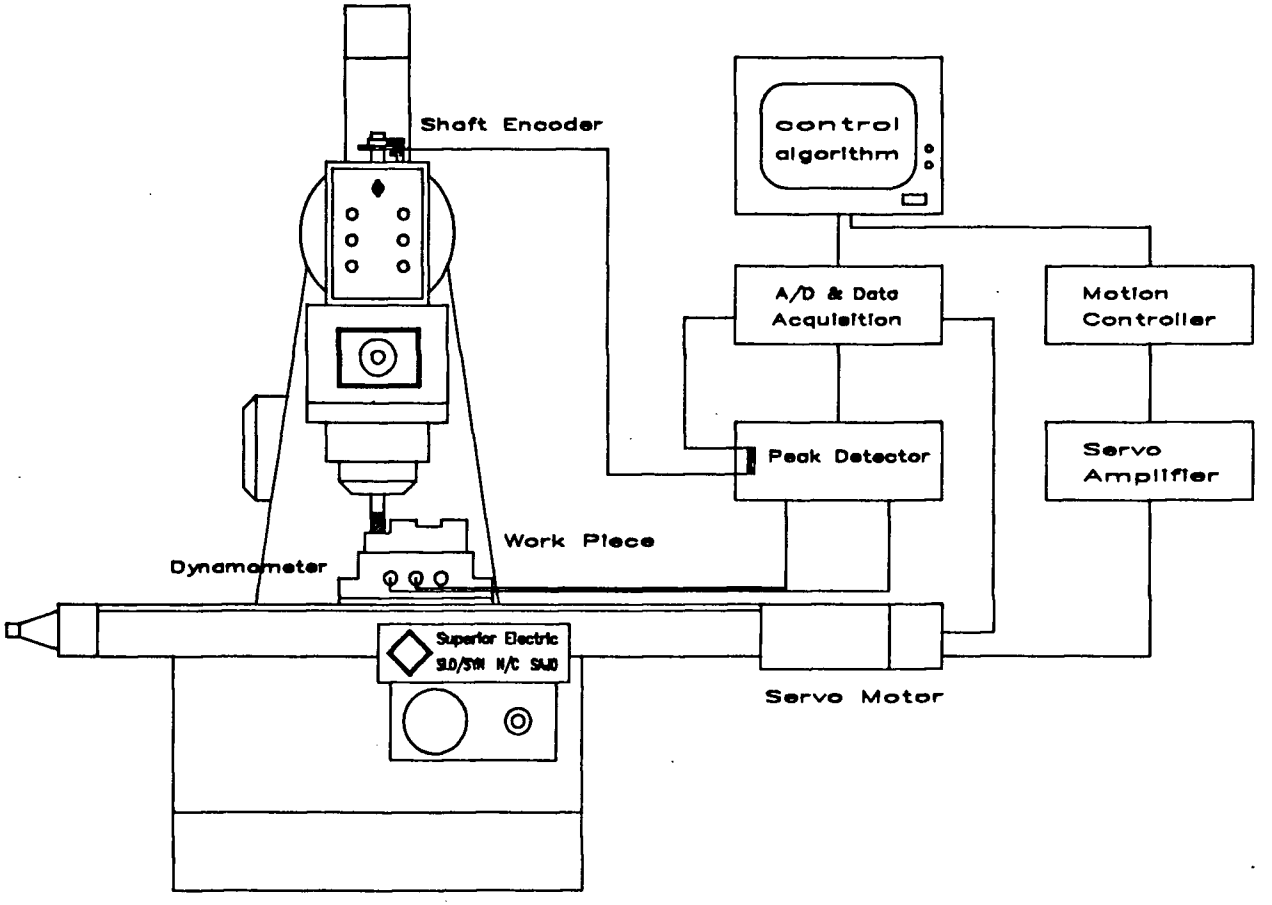

Figure 3.7: Schematic View of Experimental Setup

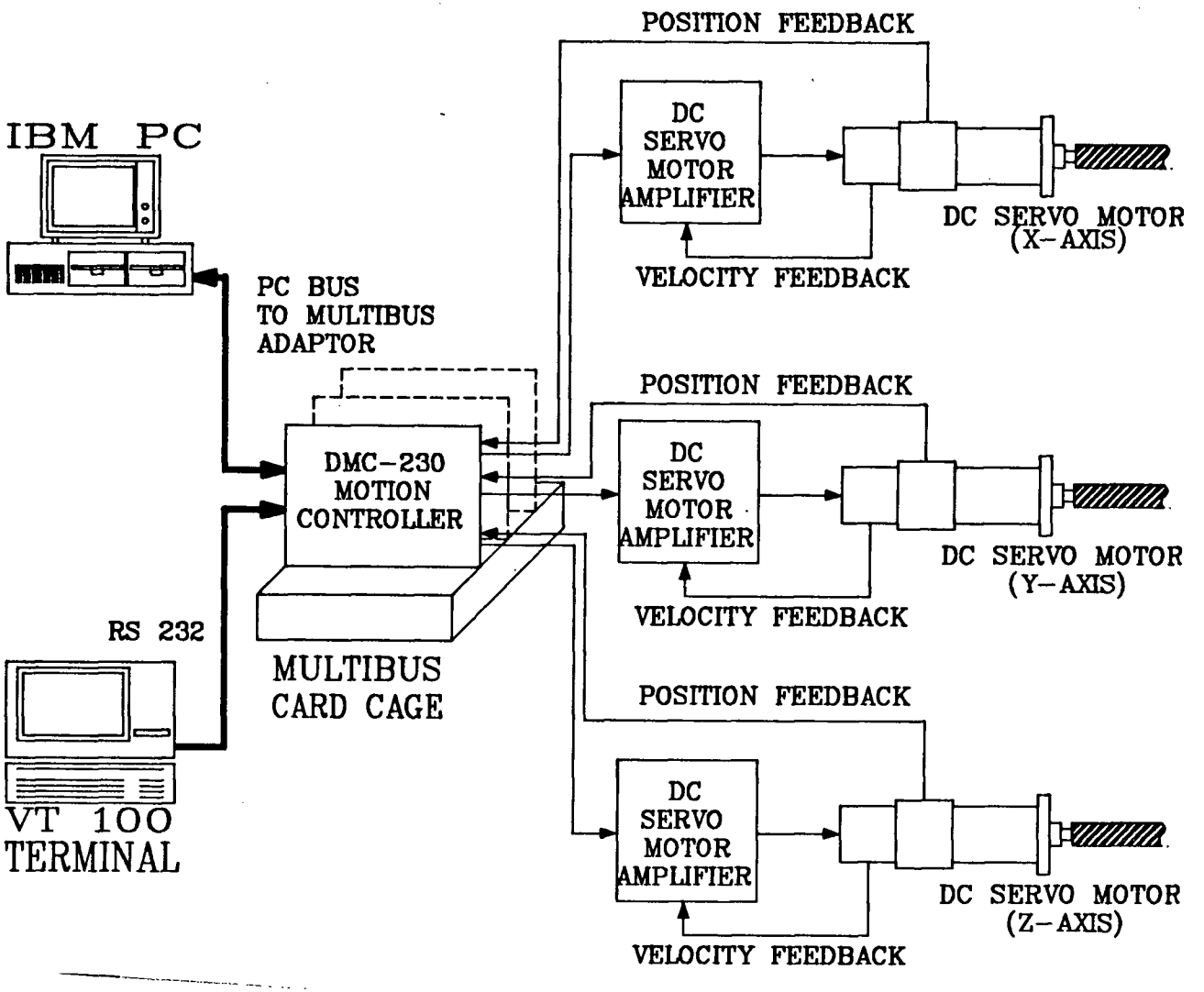

Figure 3.8: Schematic View of Servomotors

resolution of servo position and is equal to one basic length unit (BLU). The encoder is mounted on the motor shaft which is directly connected to the lead screw shaft. The lead screw has a pitch of 0.2 inches per revolution therefore every 0.001 inch linear movement of the table is equal to twenty encoder counts.

Signals to the servomotors are sent through UNIDRIVE amplifiers. The amplifiers receive tachometer feed-back signals for accurate speed control. A fully programmable three axis servomotor controller (DMC 230) is used to control positions of the three linear axis of the machine [30]. The motion control card has a Motorola 68008 Processor and 16 Input/Output (I/O) lines for logic control functions. The controller has the encoder feed-back input and a digitally controlled lead-lag compensator with a real time adjustable parameters. To send the commands to the motors, a VT 100 terminal is connected to the motion controller via an RS 232 link. For control purposes ,an Intel 80286 Central Processing Unit and a 80287 math co-processor based personal computer (PC)is used. The PC is fitted with an interface adapter ULTRALINK PC to a multibus card Cage which holds the DMC-230 controller. The controller was developed and retrofitted at UBC [29]

A three component piezo-electric dynamometer(Kistler Model 9257a) is mounted on the table to measure the cutting forces in the longitudinal and transverse directions (X and Y). On the dynamometer for each force component a proportional electrical charge is set up. These charges are sent into charge amplifiers (Dual mode model 5004) where they are converted into proportional voltages. The dynamometer has a large bandwidth with natural frequency of over 4 KHz for each axis which is much higher than the cutting force frequency of 25 Hz used in the cutting experiment. Thus the dynamics of the dynamometer could be considered as a gain.

The data acquisition system consists of two primary parts; hardware and software. The hardware includes the PC and data acquisition card model Data Translation DT

2801-A with an external connector box. An existing real time data collection software was modified at the laboratory to handle the speed of data collection. The data acquisition board can sample 8 double ended analog channels with a range of +/- 10 volts and 12 bit resolution. Data are transferred to memory via a Direct Memory Access unit, therefore the control program can continue execution independent of the data acquisition process.

In order to synchronize sampling with tooth passing frequency an external trigger is used. The external trigger is an optical encoder with a disk mounted on the spindle. The disk has a number of slots which by passing through the optical sensor send trigger signals to the data acquisition board. Two disks were used in the experiments. The first disk had 64 slots and was used in data collection for off-line analysis. For cutting process control a disk with two slots was used. The reason was that in the entire work a two teeth carbide cutter was used and the peak force at every tooth period (sample time) was needed in the control algorithms.

The force signals in X and Y directions were first sent to the peak detector box where they were squared and added together(Figure 3.9). Then the peak value at each tooth period was collected and sent along with the tacho-generator signal to the data acquisition board at each tooth period. The force signal was square rooted in software to obtain the maximum resultant force. Both tacho and force signals were multiplied by their appropriate gains to convert them from measured voltage to real values. This sampled data was processed according to the control algorithm and the controlled input signal, the feeding velocity, was sent at every sample time to the motor controller DMC 230.

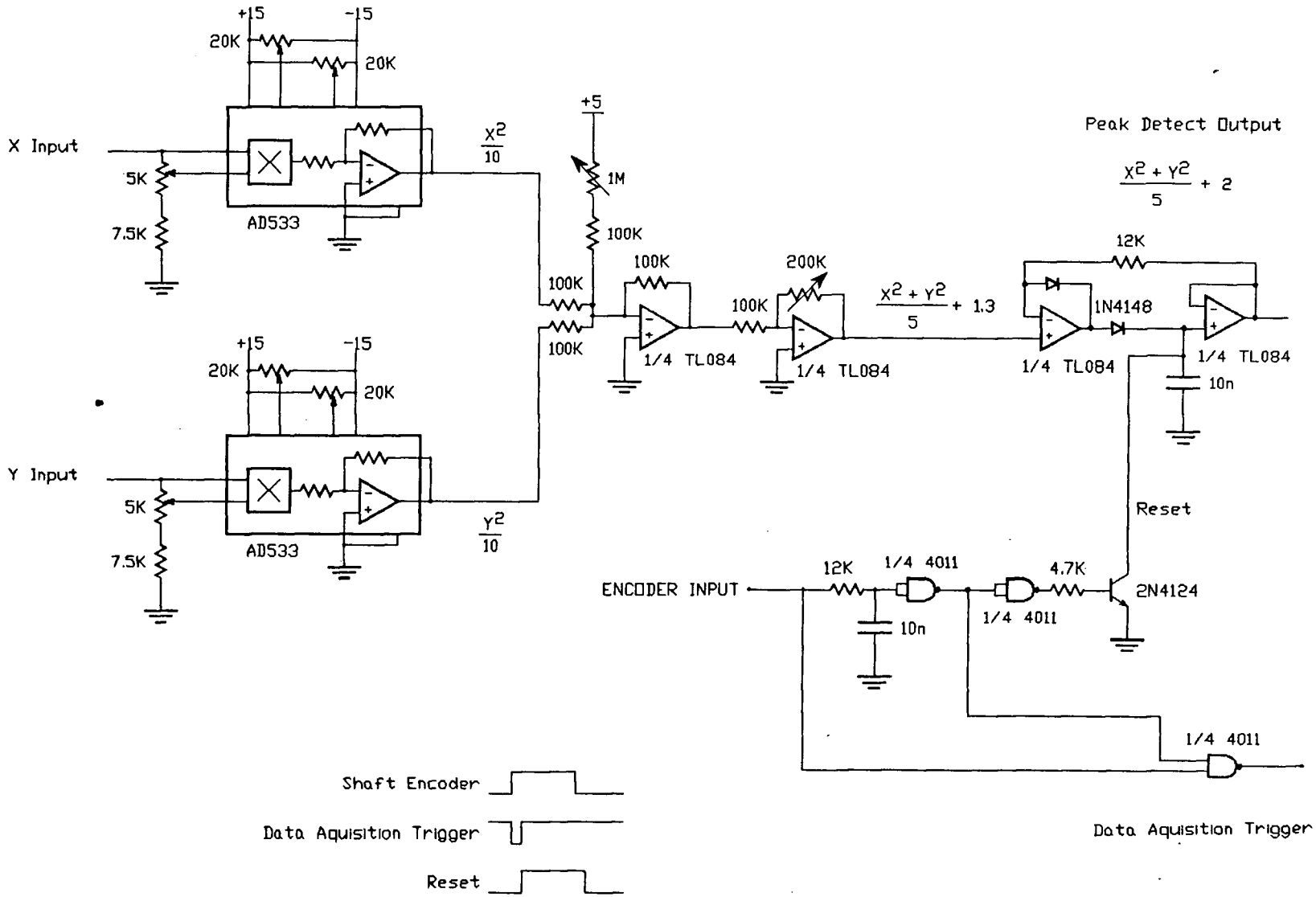

Figure 3.9: General Layout of Peak Detector

3.5 Experimental Verification of the Milling Process Modelling

An experiment was carried out in order to verify the developed model. Work-piece material was Aluminum 7075T651. The cutter was one inch in diameter with two carbide teeth and 5 degrees negative rake angle. Spindle speed and axial depth of cut(ADC) were 775 rev/min and 2.54 mm respectively. The feed-rate was varied during a full immersion milling and the cutting forces in X and Y directions, and tacho-generator signals were measured. The measurements were collected by the data acquisition system using a 64 slot shaft encoder, resulting in a sampling period of 1.21 ms. The peak resultant force at each tooth period (32 samples) is calculated from measured forces in X and Y directions. Note that the component of the cutting force in the Z direction (perpendicular to the table) is negligible for the cutter used and it is not measured. Measured peak resultant cutting force history is shown in figure3.10.

To simulate the model presented by Eq. 3.22 it is necessary to calculate the cutting constants K_s and r_1 from the above measurements. One way to calculate these parameters is to derive average forces per tooth period in feeding and normal or X and Y directions using equations 3.9 , 3.10 and 3.6 [32]

$$F_{ax} = \frac{1}{\phi_T}\int_0^{\phi_s} F_x d\phi \tag{3.33}$$

$$F_{ay} = \frac{1}{\phi_T}\int_0^{\phi_s} F_y d\phi \tag{3.34}$$

or

$$F_{ax} = \frac{K_s\, a\, s_t}{\phi_T}\int_0^{\phi_s} (\sin\phi\cos\phi + r_1\sin^2\phi)d\phi \tag{3.35}$$

$$F_{ay} = \frac{K_s\, a\, s_t}{\phi_T}\int_0^{\phi_s} (\sin^2\phi - r_1\,\sin\phi\cos\phi)d\phi \tag{3.36}$$

where

Z the number of teeth on the cutter

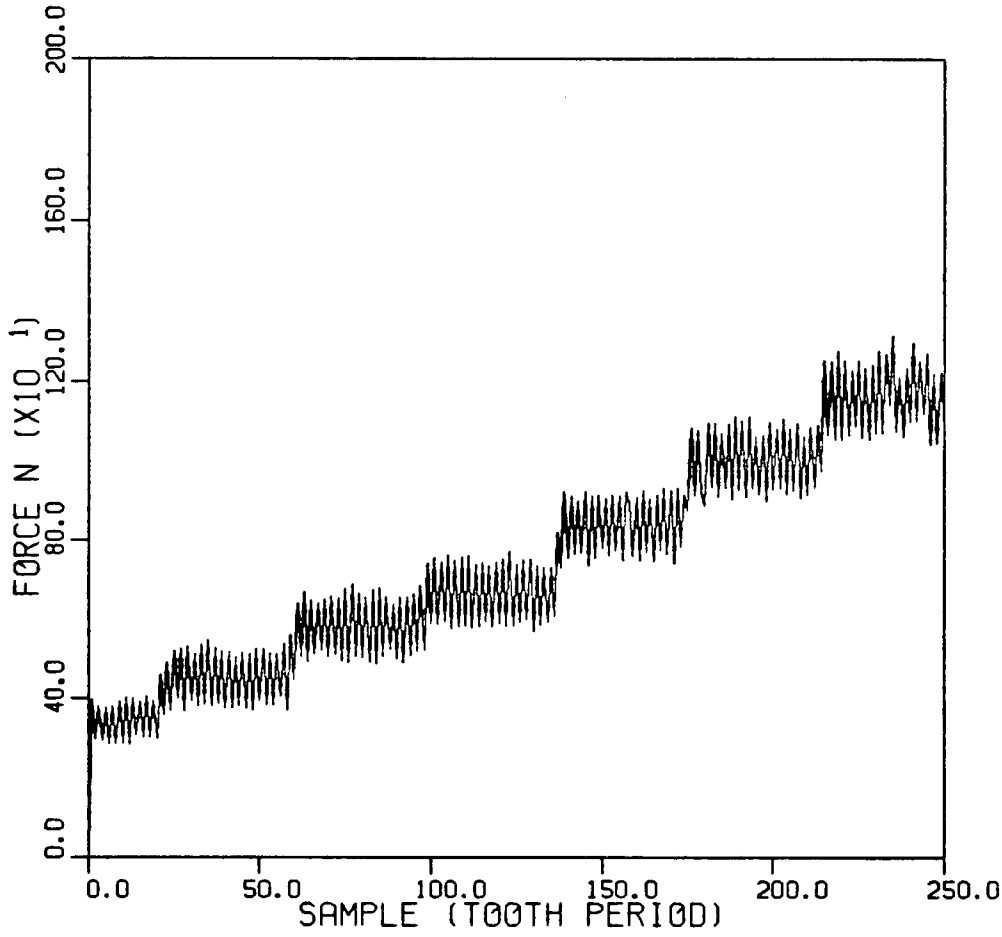

Figure 3.10: Measured Peak Resultant Cutting Force for 2.54 mm Two Teeth Carbide Cutter, Axial Depth of Cut 2.54 mm, Spindle Speed 775 RPM, Feed-rates .024, .049, .073, .098, .147, .196 , .245 mm/rev/tooth

$\phi_T = \frac{2\pi}{Z}$ cutter pitch angle

ϕ_S swept angle of cut

It is assumed that there is no run-out and the tooth spacing is uniform. Integrating the above equations

$$F_{ax} = \frac{K_s\, a\, s_t\, Z}{8\,\pi}\{(1-\cos 2\phi_S) + r_1(2\phi_S - \sin 2\phi_S)\} \tag{3.37}$$

$$F_{ay} = \frac{K_s\, a\, s_t\, Z}{8\,\pi}\{(2\phi_S - \sin 2\phi_S) - r_1(1-\cos 2\phi_S)\} \tag{3.38}$$

For two teeth cutter and full immersion one gets

$$F_{ax} = \frac{r_1\, K_s\, a\, s_t}{2} \tag{3.39}$$

$$F_{ay} = \frac{K_s\, a\, s_t}{2} \tag{3.40}$$

Average forces in the X and Y directions in every tooth period were calculated from measurements and plotted versus the feed-rate. The results are shown in figure3.11. Note that the curves do not pass the origin due to presence of edge forces at low feed-rates.

The cutting constants are calculated from above equations and figure3.11

K_s =1212 N/mm^2

r_1 =.78

The equivalent stiffness of the tool on the spindle, K_x ,was measured statically. A proximeter sensor was used to measure deflection of the tool in response to the horizontal loads applied to the tool. Deflection of the spindle was measured separately and was negligible. Figure 3.12 shows deflection of the cutter under the load. The value of K_x is equal to 12100 N/mm.

The model of milling process (Eq.3.22) is simulated with the above cutting constants and the result is shown in Figure3.13. The effect of run-out is simulated and

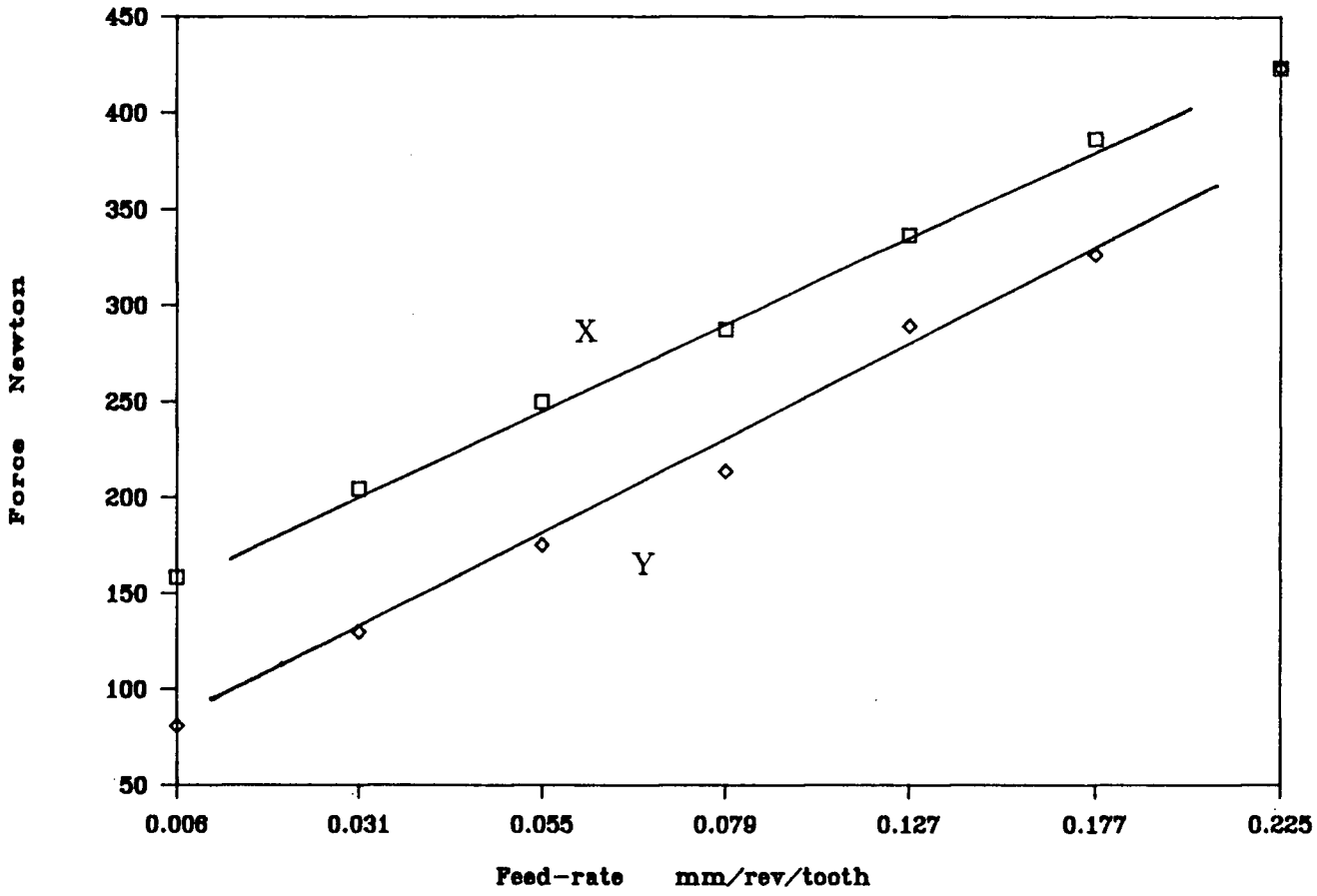

Figure 3.11: Measured Average Cutting Forces in X and Y Directions for One Inch Two Teeth Carbide Cutter, Axial Depth of Cut 2.54 mm, Spindle Speed 775 RPM

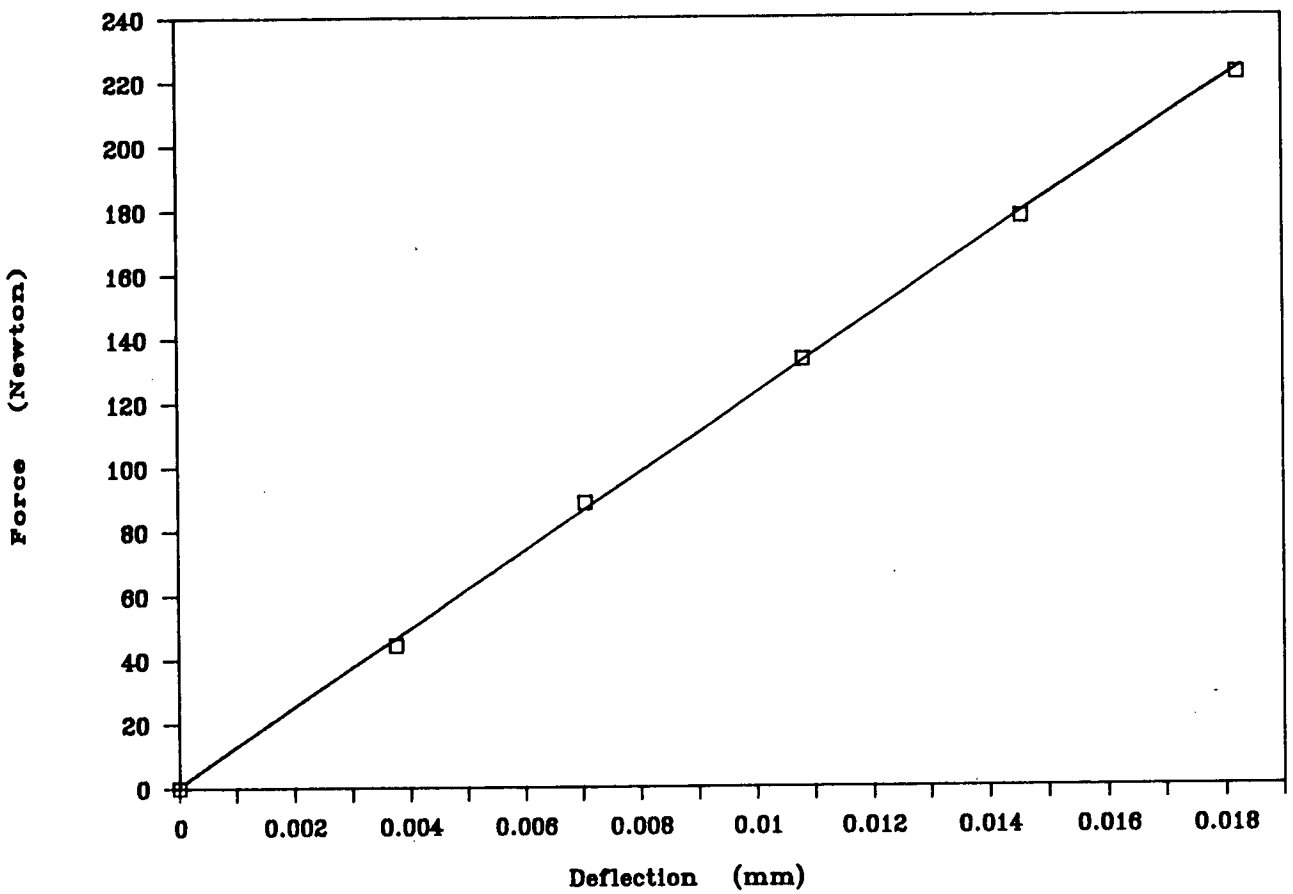

Figure 3.12: Deflection of One Inch Two Teeth Carbide Cutter Mounted on the Spindle under Varying Load

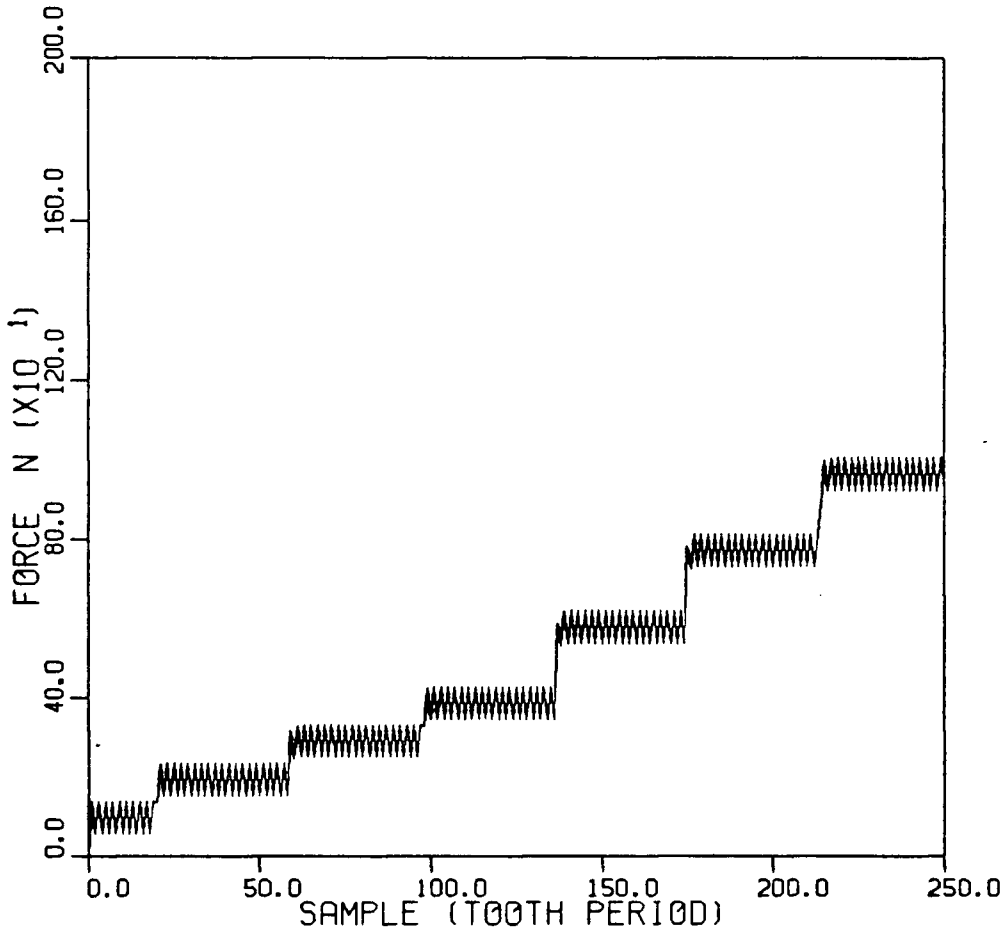

Figure 3.13: Simulated Peak Resultant Cutting Force for One Inch Two Teeth Carbide Cutter, Axial Depth of Cut 2.54 mm, Feed-rates .024, .049, .073, .098, .147, .196, .245 mm/rev/tooth

superimposed on the peak force. Comparing Figs. 3.10 and 3.13 which have identical conditions shows that the model is following the same trend as the experiment. The DC offset in the measured value is not crucial in the control design because the parameters of the model are estimated on line in the control loop. This point is further clarified in the next experiment.

It was stated previously that the cutting process model is first order but under special conditions it could be treated as a gain. In another experiment the feed-rate was changed alternatively between .295 and .59 mm/rev/tooth while the axial depth of cut was kept constant (2.54 mm). Feed-rate was changed after 10 spindle revolutions. In order to completely get rid of the effect of run-out a cutter with one tooth was used. The spindle speed was 362 RPM. Both tacho and force signals were passed through a 100 Hz low pass filter. The history of measured peak force is shown in Fig.3.14. The transients of the measured peak force show the dynamics of milling forces which is different from a simple gain. A first order model can follow the transients and is simulated according to Eq. 3.22 with the same cutting constants calculated above. The result is shown in Fig.3.14.

In control algorithms the tacho and peak resultant force signals are used for identification. Here the cutting process model is treated as a gain and also a first order system and these signals are used to build the predicted output. Figure 3.15 shows the measured peak resultant force along with predicted forces for gain and first order systems. The first order system is following the measured force better than the gain. The cost function, based on summation of squares of errors, is also lower for the first order system.

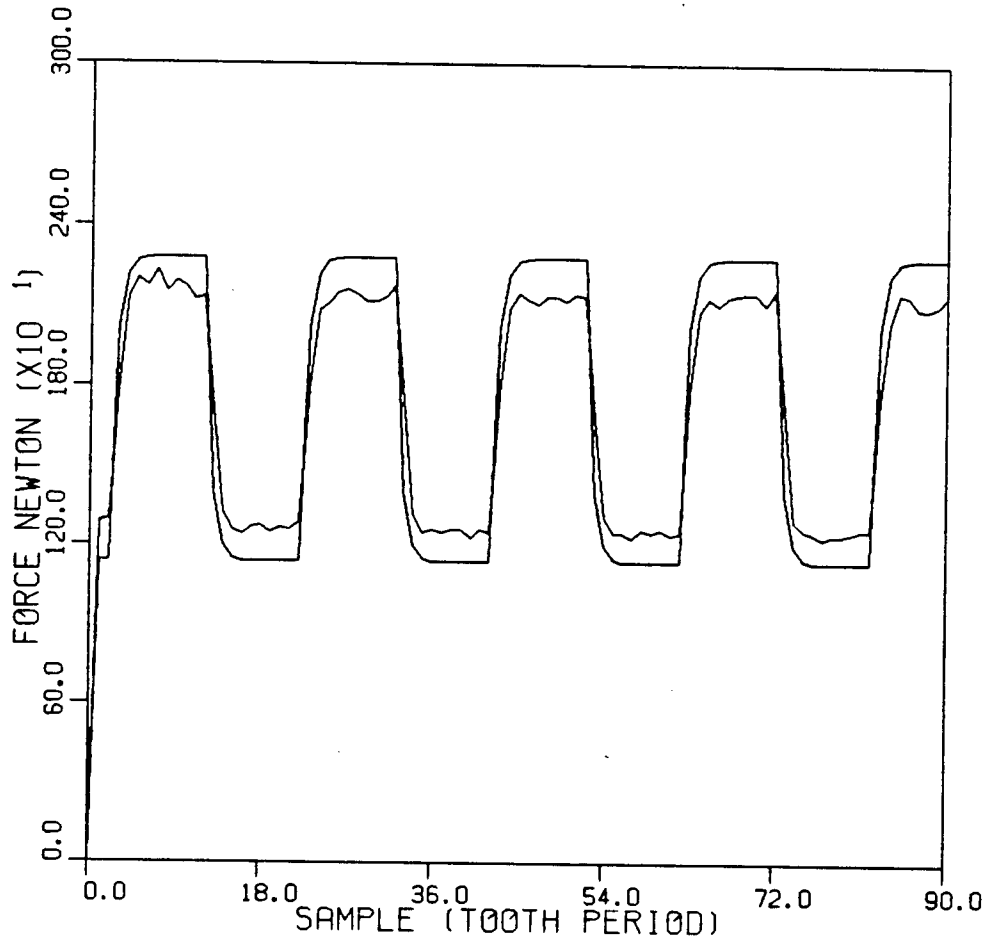

Figure 3.14: Measured Peak Resultant Force and Simulated First Order Model Feed-rates .295 and .59 mm/rev/tooth, Axial Depth of Cut 2.54 mm, Spindle Speed 362 RPM

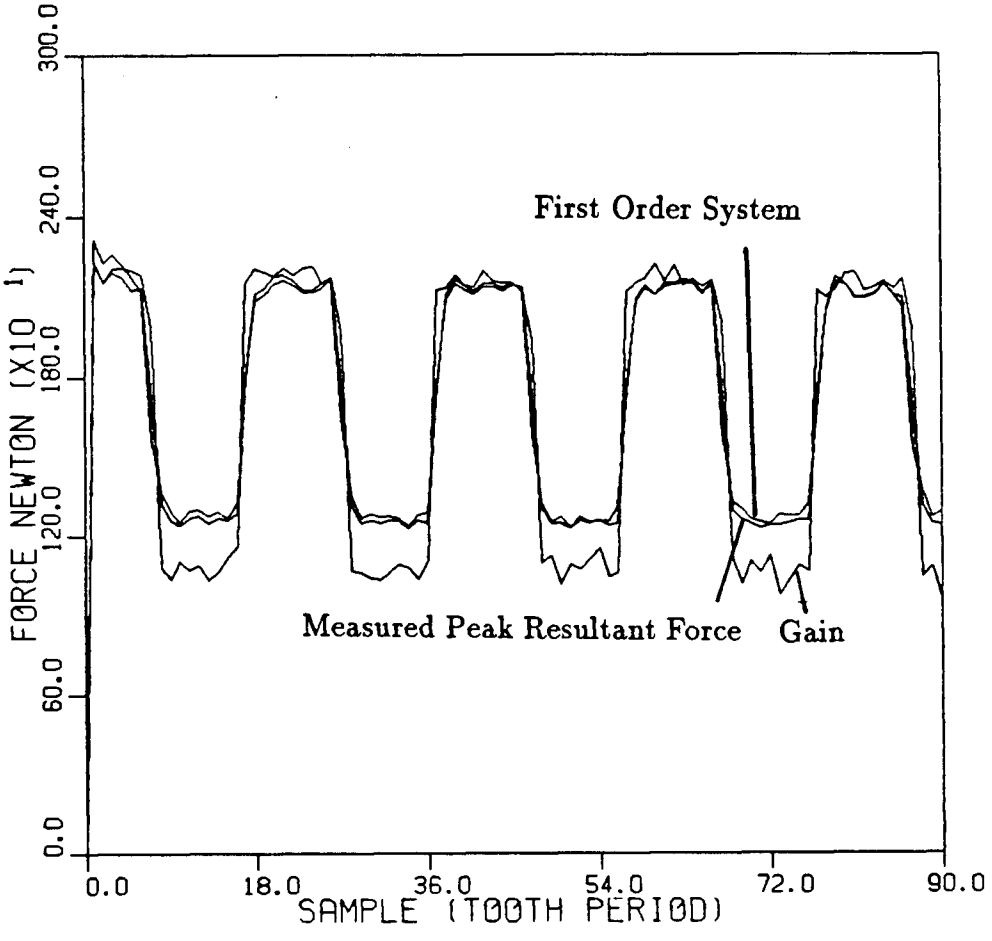

Figure 3.15: Measured Peak Resultant Force and Predicted Forces
a) Measured Peak Resultant Force
b) Predicted Force for the Cutting Process as a Gain
c) Predicted Force for the Cutting Process as a First Order System
Feed-rates .295 and .59 mm/rev/tooth, Axial Depth of Cut 2.54 mm, spindle speed 362 RPM

Chapter 4

Controller Design and Application

4.1 Introduction

The adaptive control loop of machining processes has three fundamental control blocks. The first part is the controller which adaptively manipulates the feed-rate in order to keep the cutting forces below a desired level. The second block is the time invariant dynamics of the feed drive position control system. The third block represents the time varying dynamics of the machining process which was modelled in the previous section.

In this chapter first the discrete time model of the feed drive control system will be derived. Since the machining process parameters are time varying the method used for their real time identification will be briefly discussed. Three adaptive control algorithms, which are used to control the milling forces are simulated and implemented on the CNC machine.

4.2 Modelling of the Feed Drive Controller

The research machine is a vertical knee type three axis milling machine, namely longitudinal, normal and vertical. Each axis has a recirculating ball screw drive controlled by a permanent magnet, pulse width modulated direct current servo motors. The motors are directly connected to the feed drive shafts without any gear reduction. The velocity and position of each axis is controlled by an analog and a digital control modules. A block diagram of the feed drive controller is shown in figure 4.16.

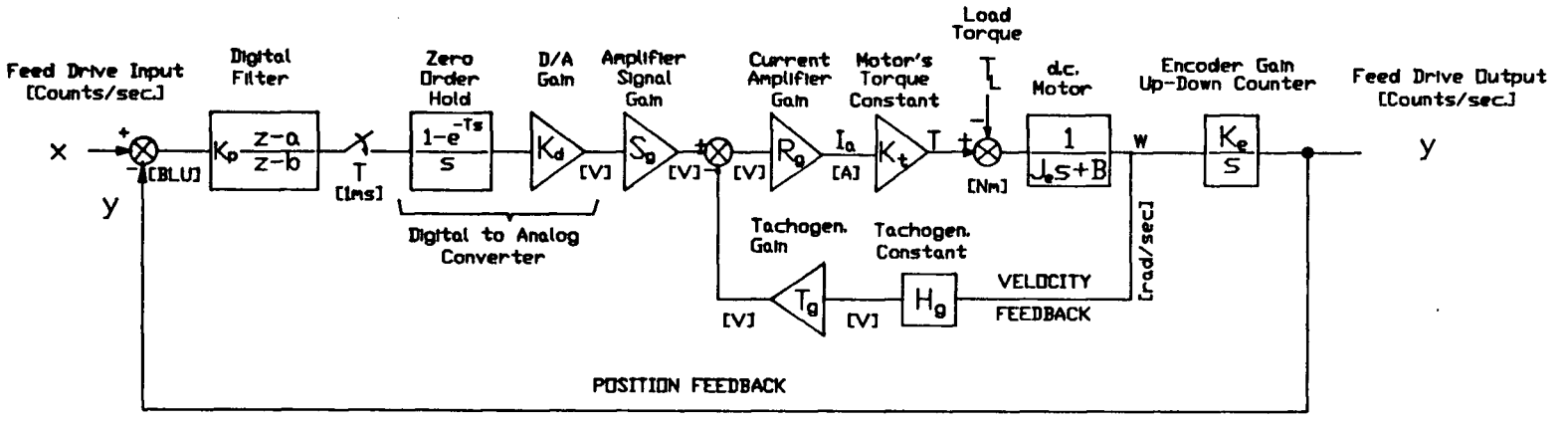

Figure 4.16: Block Diagram of Feed Drive Controller

The analog part of the system consists of an analog amplifier card and a motor. The amplifier card has a velocity feed-back signal from tacho-generator with tacho constant H_g and tacho gain T_g , amplifier signal gain S_g, current amplifier gain R_g and motor torque constant K_t. Motor consists of the dc motor's first order transfer function and up-down counter encoder gain K_e. The transfer function of the continuous part is calculated as follows

From the feed forward part of the velocity feed-back loop one gets

$$G_1(s) = R_g\, K_t\, \frac{1}{J\, s\, + B_f} \tag{4.41}$$

where J is equivalent inertia reflected to the motor shaft and B_f is friction coefficient. Then the closed loop transfer function of the velocity feed-back loop is calculated as

$$G_2(s) = \frac{G_1}{1 + G_1\, H1} = \frac{R_g\, K_t}{J\, s\, +\, B_f\, +\, R_g\, K_t\, T_g\, H_g} \tag{4.42}$$

Note that load torque is set equal to zero. Combining the above equation with the D/A gain, amplifier gain and encoder gain results in

$$G_3(s) = G_2(s)\, \frac{K_{da}\, S_g\, K_e}{s} \tag{4.43}$$

or

$$G_3(s) = \frac{K_d\, S_g\, R_g\, K_t\, K_e/J}{s\, [s\, +\, (B_f\, +\, R_g\, K_t\, T_g\, H_g)/J]} \tag{4.44}$$

The digital part consists of a digital filter with gain K_P , zero order hold, D/A gain K_{da} and position feed-back signal from encoder. The digital filter has transfer function

$$G_4(z) = K_p \frac{z\, -\, a}{z\, -\, b} \tag{4.45}$$

The sampling period of the adaptive control is equal to the tooth period which is far greater than the sampling time of the digital filter T_1. Thus using Tustin transformation

$$z = (2\, +\, s\, T_1)/(2\, -\, s\, T_1) \tag{4.46}$$

and setting $b = 0$ results in

$$G_4(z) = K_p \frac{s + \frac{2-2a}{T_1 + a\,T_1}}{\frac{2 + s\,T_1}{T + a\,T_1}} \tag{4.47}$$

The CNC system is tuned such that the zero of the digital filter cancels the velocity feed-back pole

$$\frac{2 - 2\,a}{T_1 + a\,T_1} = \frac{B_f + R_g\,K_t\,T_g\,H_g}{J} \tag{4.48}$$

multiplying together

$$G_5(s) = G_3(s)\,G_4(s) = K_p \frac{(T_1 + a\,T_1)}{(2 + s\,T_1)} \frac{(K_{da}\,S_g\,R_g\,K_t\,K_e)}{(s\,J)} \tag{4.49}$$

The closed loop transfer function with encoder feed-back is

$$G_S(s) = \frac{G_5}{1 + G_5} \tag{4.50}$$

or

$$G_S(s) = \frac{[(1 + a)\,K_{da}\,S_g\,R_g\,K_t\,K_e\,K_p]/J}{s^2 + (2\,s)(T_1) + [(1 + a)\,K_{da}\,S_g\,R_g\,K_t\,K_e]/J} \tag{4.51}$$

where

J=0.006	Kg m^2
K_d =0.0781	V/Count
S_g=0.0627	V/V
R_g=208.3	A/V
K_t=0.3	N m/A
K_e =636.6	Counts/rad
T_1=0.001	sec

From equation 4.48 one gets $a = 0.882$ and K_p is designed as $K_p = 2.5$ to obtain a damped position response. Substitution of the parameters into equation 4.51 yields

$$G_S(s) = \frac{152591.6}{s^2 + 2000\,s + 152591.6} \tag{4.52}$$

Therefore a second order transfer function is obtained for the servo. The natural frequency and damping ratio of the servo feeding velocity are 390.67 rad/sec and 2.56 respectively. Readers are referred to [34] for more information regarding the dynamics of the feed drive controller.

4.3 Parameter Identification Method

There are several methods to estimate the parameters of a time varying linear process. The recursive least square technique (RLS) is most popular in control problems and is briefly discussed here. A detailed description of the identification methods can be found in [35].

Consider a dynamic system with the sampled input signal $u(t)$ and output signal $y(t)$. The sampled values can be related through a linear difference equation

$$A(z^{-1})\, y(t) \;=\; B(z^{-1})\, u(t) \tag{4.53}$$

where $A(z^{-1})$ and $B(z^{-1})$ are polynomials defined in backward shift operator (z^{-1})

$$A(z^{-1}) \;=\; 1 \;+\; a_1 z^{-1} \;+\cdots+\; a_n z^{-n} \tag{4.54}$$

$$B(z^{-1}) \;=\; b_1 z^{-1} \;+\; b_2 z^{-2} \;+\cdots+\; b_m z^{-m} \tag{4.55}$$

n and m are the order of the polynomials A and B respectively. The above model could be written in a concise form as

$$y(t) \;=\; \theta^T\, \phi(t) \tag{4.56}$$

where

$$\theta^T \;= (a_1, \ldots, a_n,\; b_1, \ldots, b_m) \tag{4.57}$$

is the parameter vector and

$$\phi^T(t) \;= (-y(t-1), \ldots, -y(t-n),\; u(t-1), \ldots, u(t-m)) \tag{4.58}$$

is the observation vector and its components are called regressors.

The parameter vector θ is to be estimated from measurements of input and output over N sampling time i.e. $y(t)$,$\phi(t)$,$t = 1, 2, \ldots, N$ and by minimizing a cost function $V_N(\theta)$

$$V_N(\theta) = \frac{1}{N} \sum_{t=1}^{N} \alpha_t \left[y(t) - \theta^T \phi(t) \right]^2 \tag{4.59}$$

The above equation represents the variance of measured and estimated output. α_t is a weighting factor for observations and often selected equal to one. This criterion is quadratic in θ therefore it can be minimized analytically with respect to θ which gives an estimated parameter vector as

$$\hat{\theta}(N) = \left[\sum_{t=1}^{n} \alpha_t \, \phi(t) \, \phi(t)^T \right]^{-1} \sum_{t=1}^{N} \alpha_t \, \phi(t) \, y(t) \tag{4.60}$$

provided that the inverse of the matrix exists.

It could be shown that[35] the estimated parameter vector could be found in recursive form at each sampling interval by

$$\hat{\theta}(t) = \hat{\theta}(t-1) + L(t) \left[y(t) - \hat{\theta^T}(t-1) \, \phi(t) \right] \tag{4.61}$$

$$L(t) = \frac{P(t-1)\phi(t)}{1/\alpha_t + \phi^T(t) \, P(t-1)\phi(t)} \tag{4.62}$$

$$P(t) = \frac{1}{\lambda} \left[P(t-1) - \frac{P(t-1)\phi(t)\phi^T(t)p(t-1)}{\lambda/\alpha_t + \phi^T(t)p(t-1)\phi(t)} \right] \tag{4.63}$$

Where λ is the discounting or forgetting factor, $L(t)$ is the gain vector and $P(t)$ is the covariance matrix. The covariance matrix and parameter vector should be initialized before start up. The choice of forgetting factor is stated in simulations. Table 4.1 presents a psuedo code for recursive least square method.

Table 4.1: Psuedo Code for Parameter Identification Method

1) Define the model of the process according to equation 4.53 i.e. order of parameters A and B

2) Setup parameter vector and covariance matrix and initialize them(equations 4.57 and 4.63). A common choice of initial values is $\hat{\theta}(0) = 0$ and $P(0) = C.I$ where C is some large constant and I is the unity matrix.

3) Update the observation vector with the new data(equation 4.58).

4) Calculate the covariance matrix (equation 4.63).

5) Calculate the gain vector(equation 4.62).

6) Calculate the new parameters(equation 4.61).

7) Return to 3 and repeat.

4.4 Adaptive PID Control Design

4.4.1 Introduction

PID control is popular in industry, because it is simple and relatively easy to implement. In PID control, a proper combination of three control actions manipulate a variable, say the error signal. To obtain the control input signal, the error is multiplied by a proportional gain, integrated , differentiated and multiplied by integral and derivative gains respectively where all three actions combined together. PID control is similar to a phase lead-lag compensator. It affects both high and low frequency regions, improves system stability, increases speed of response and reduces steady-state error[33]. As long as the plant parameters do not change, the gains of a PID controller could be set for a stable operation. For time varying processes these parameters should be calculated adaptively. Although PID control is popular its implementation on machining processes has not been yet reported.

4.4.2 Adaptive PID Design for Milling Process Control

Here a self-tuning structure is used as shown in figure 4.17. In general the parameters of the plant should be identified at every sampling time and the controller parameters adjusted according to the design procedure. There are a couple of problems associated with this approach, namely convergence of the parameters and computation time necessary to identify the parameters of the process. One way to overcome this problem is to decouple the time invariant feed-drive servo from the time varying cutting process block. Therefore only the parameters of the cutting process are identified from feed-rate and force measurements, using RLS, and then those parameters are used to determine the PID gains as will be explained later. This technique will decrease the number of estimated parameters from five to two and not only reduces the computation

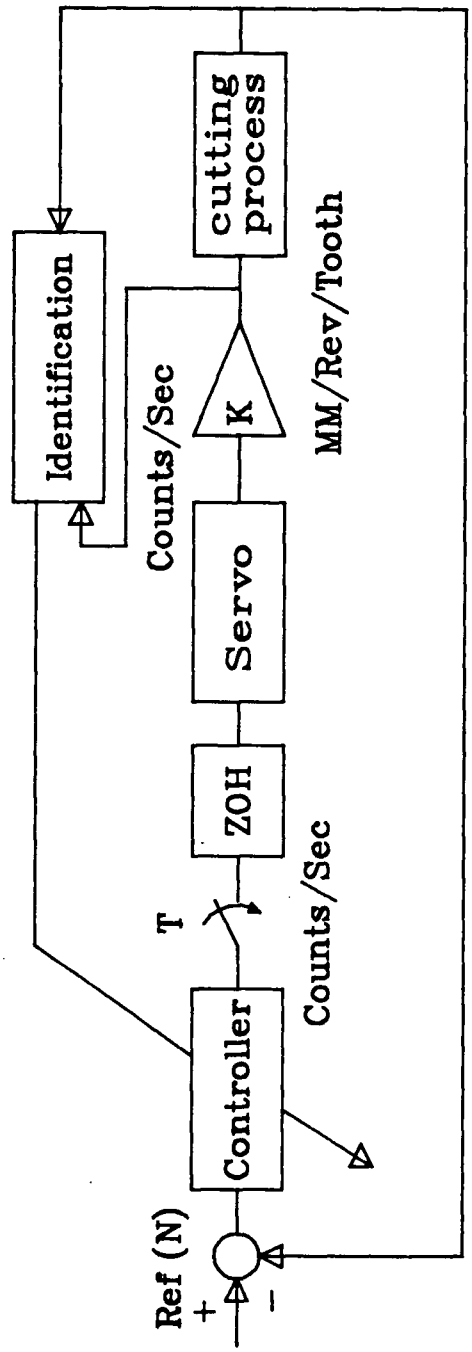

Figure 4.17: Adaptive PID Control Loop

time, which is crucial in real time applications, but also increases the accuracy of the identification for finding the control input.

The choice of sampling time is important in control design. High sampling rates are favorable but then the cost of hardware increases. On the other hand, low sampling rates would corrupt the control input and may result in instability [37,33,38]. Sampling time should be small but greater than the time required to identify the parameters and calculate the control input. In this work sampling time of one tooth period is chosen. The reason is that cutting forces in milling are periodic with tooth passing frequency and the most important information is the peak resultant force at each tooth period. A two teeth cutter was used in all experiments shown in this thesis. Spindle speed was 775 RPM in control design. Therefore the sampling period is calculated as

$$T = \frac{60}{N\ Z} = \frac{60}{(2)(775)} = 0.0387 \tag{4.64}$$

Considering the available computation power , this sampling time is sufficient to estimate the parameters and calculate the control input.

The zero order hold equivalence of the servo with sampling time equal to one tooth period is

$$G_S(z) = \frac{0.9518086(z + 0.0020946)}{z(z - 0.0461992)} \tag{4.65}$$

The cutting process was shown to have the structure of

$$G_P(z) = \frac{b_0}{z - a_1} \tag{4.66}$$

The structure of the PID controller is[33]

$$G_C(z) = K_p + \frac{K_I}{1 - z^{-1}} + K_d\,(1 - z^{-1}) \tag{4.67}$$

or

$$G_C(z) = \frac{z^2\,(K_p + K_I + K_d\,) + z\,(-K_p - 2\,K_d\,) + K_d}{z(z - 1)} \tag{4.68}$$

Three unknown parameters of the controller (i.e. K_P, K_I and K_d) are to be identified at each tooth period.

The parameters of the feed drive servo are time invariant and are given in equation 4.65. The feed drive servo has two poles, one at the center (z=0) and the other is close to the center (i.e. z=0.0461992). The second pole can be safely canceled by one of the zeros of the PID controller. The cutting parameters are estimated recursively at each tooth period. It was shown in the last chapter that the parameter a_1 is always less than one in the milling process, thus the estimated pole of the process is expected to be within the unit circle and it can safely be canceled by another zero of the PID controller.

The coupled transfer function of the feed drive servo and the cutting process can be defined as plant $G_{PI}(z)$

$$G_{PI}(z) = G_S(z)G_P(z) = \frac{0.9518086(z + 0.0020946)K\hat{b}_0}{z[z^2 - (0.0461992 + \hat{a}_1)z + (0.0461992)\hat{a}_1]} \tag{4.69}$$

Where K is the gain of the table's leadscrew and equal to 0.0000491 mm/rev/tooth/counts/sec. $\hat{b}_0$ and $\hat{a}_1$ are the estimates of the cutting process parameters. To have pole-zero cancelations

$$z^2 - \frac{K_P + K_d}{K_P + K_I + K_d}z + \frac{K_d}{K_P + K_I + K_d} \equiv z^2 - (0.0461992 + \hat{a}_1)z + (0.0461192)\hat{a}_1 \tag{4.70}$$

The polynomial coefficients must be equal to satisfy the above equation

$$\frac{K_p + 2K_d}{K_p + K_I + K_d} = \hat{a}_1 + 0.0461992 \tag{4.71}$$

$$\frac{K_d}{K_p + K_I + K_d} = \hat{a}_1 (0.0461992) \tag{4.72}$$

There are three unknown parameters with two equations. The third necessary equation can be found from the stability requirement of the system. Jury's stability test shows

whether the roots of a polynomial are inside the unit circle. The procedure is simple to apply and can be found in most control text books. To apply jury's test the characteristic equation should be derived. Assuming that the two polynomials are canceled due to pole-zero cancelation, the remaining open loop transfer function of the system reduces to

$$G_{OL}(z) = \frac{K\,(K_p + K_I + K_d)\,b_0\,(0.9518086)\,(z + .0020946)}{z^2(z - 1)} \tag{4.73}$$

The closed loop transfer function of the system becomes

$$G_{CL} = \frac{G_{OL}(z)}{1 + G_{OL}(z)} \tag{4.74}$$

$$G_{CL} = \frac{A(z + 0.0020946)}{z^3 - z^2 + Az + (0.0020946)A} \tag{4.75}$$

where

$$A = K(K_P + K_I + K_d)(0.9518086)\hat{b}_0 \tag{4.76}$$

The characteristic equation is

$$F(z) = z^3 - z^2 + A\,z + .0020946\,A \tag{4.77}$$

or

$$\begin{aligned} F(z) = {} & z^3 - z^2 + (K_P + K_I + K_d)(K\hat{b}_0\,0.9518086) \\ & + (K_P + K_I + K_d)(K\hat{b}_0)(0.0020946)(.9518086) \end{aligned} \tag{4.78}$$

The general form of characteristic equation is

$$F(z) = a_0 z^n + a_1 z^{n-1} + \cdots + a_{n-1} z + a_n \tag{4.79}$$

The requirements for Jury's test are as follows

1) $| a_n | < 1$ which means

$$0.0020946\,A < 1 \tag{4.80}$$

or

$$K(K_P + K_I + K_d)(0.0019936) < 1 \tag{4.81}$$

2) $F(z = 1) > 0$ or

$$A + 0.0020946A > 0 \tag{4.82}$$

and because both lead screw gain K and cutting process gain $\hat{b}_0$ are positive results to

$$K_P + K_I + K_d > 0 \tag{4.83}$$

3) $F(z = -1) < 0$

$$-2 - A + 0.0020946A < 0 \tag{4.84}$$

which does not give any new information

4) From Jury's table

$$|\ (0.0020946A^2\ - 1\ |>|\ 0.0020946A\ + A\ | \tag{4.85}$$

which with the help of the first condition and neglecting the second order term reduces to

$$A < 0.9979097 \tag{4.86}$$

or

$$K_P + K_I + K_d < \frac{1.0484353}{\hat{b}_0 K} \tag{4.87}$$

Equations 4.83 and 4.87 show the limits of stability of PID controller. However, these equations do not indicate where the poles of the closed loop system are located. The two conditions can be unified as

$$0\ <\ K_P + K_I + K_d\ <\ \frac{1.0484353\ K_G}{\hat{b}_0 K} \tag{4.88}$$

where K_G is a factor between zero and one.

To have a better insight about the location of the poles, equation4.88 is inserted in equation4.73 and the root locus of the system is plotted as shown in figure4.18. One pole always remains close to the centre of the unit circle. As K_G increases, the two poles one from origin and the other on the unit circle move towards each other on real axis. When the poles coincide K_G has a value of 0.2495 and the system becomes critically damped. The poles branch from the real axis and reach to their final values of the $0.5019 \pm .8618j$ on the unit circle where the system reaches the margin of stability and K_G becomes unity. In this work K_G is selected as 35 % of its final value, which corresponds to one pole at the origin and two others at $0.5019 \pm 0.3158j$. The system has a damping ratio of 0.71 and natural frequency of 20.3 rad/sec. The rise time and settling time for the system are 123 millisec and 323 millisec respectively. Classically this is the optimum choice because with a small overshoot the fastest response is achieved.

Table4.2 shows a psuedo code for PID control. algorithm

4.4.3 Simulation and Experimental Results

This controller is simulated for varying axial depth of cut to evaluate the performance of the system. The cutting pattern is shown in figure4.19. The same work-piece profile was used for all the experiments. Figure4.20 shows the cutting force and control input respectively. The controller adjusts to the feed rate such that after each step change in axial depth of cut the resultant force tracks the required reference force. The jumps in the cutting force are due to the time required to identify plant parameters after sudden changes in axial depth of cut. Identification plays a major rule in performance of this type of controllers. Ideally the time required to identify the parameters is at least equal to the number of parameters multiply by the sampling time (which in this case is one revolution of the cutter). In practice it is larger and relates to the richness of input excitation. The forgetting factor is set equal to one. When the difference between the

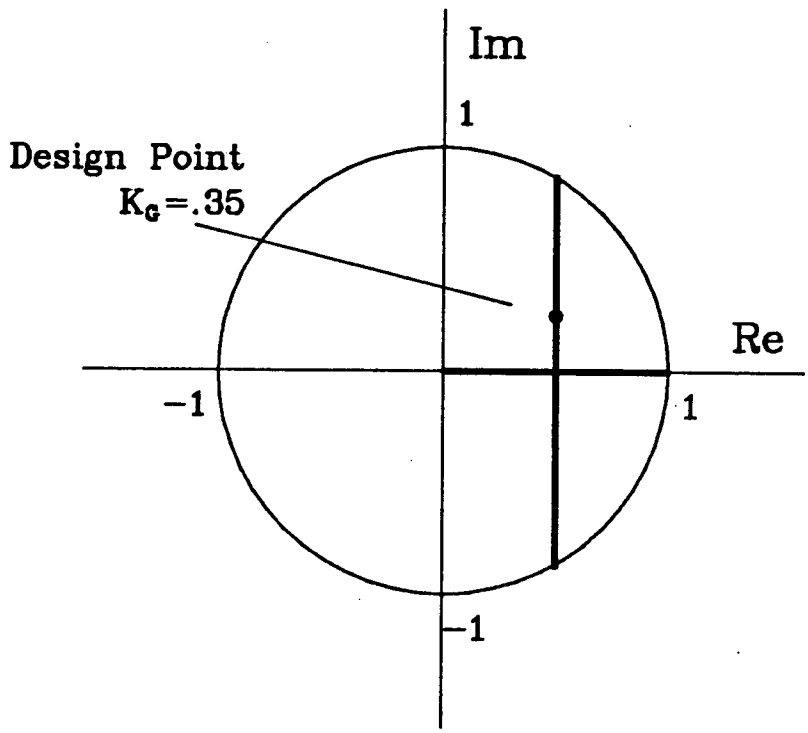

Figure 4.18: Root Locus for PID Design

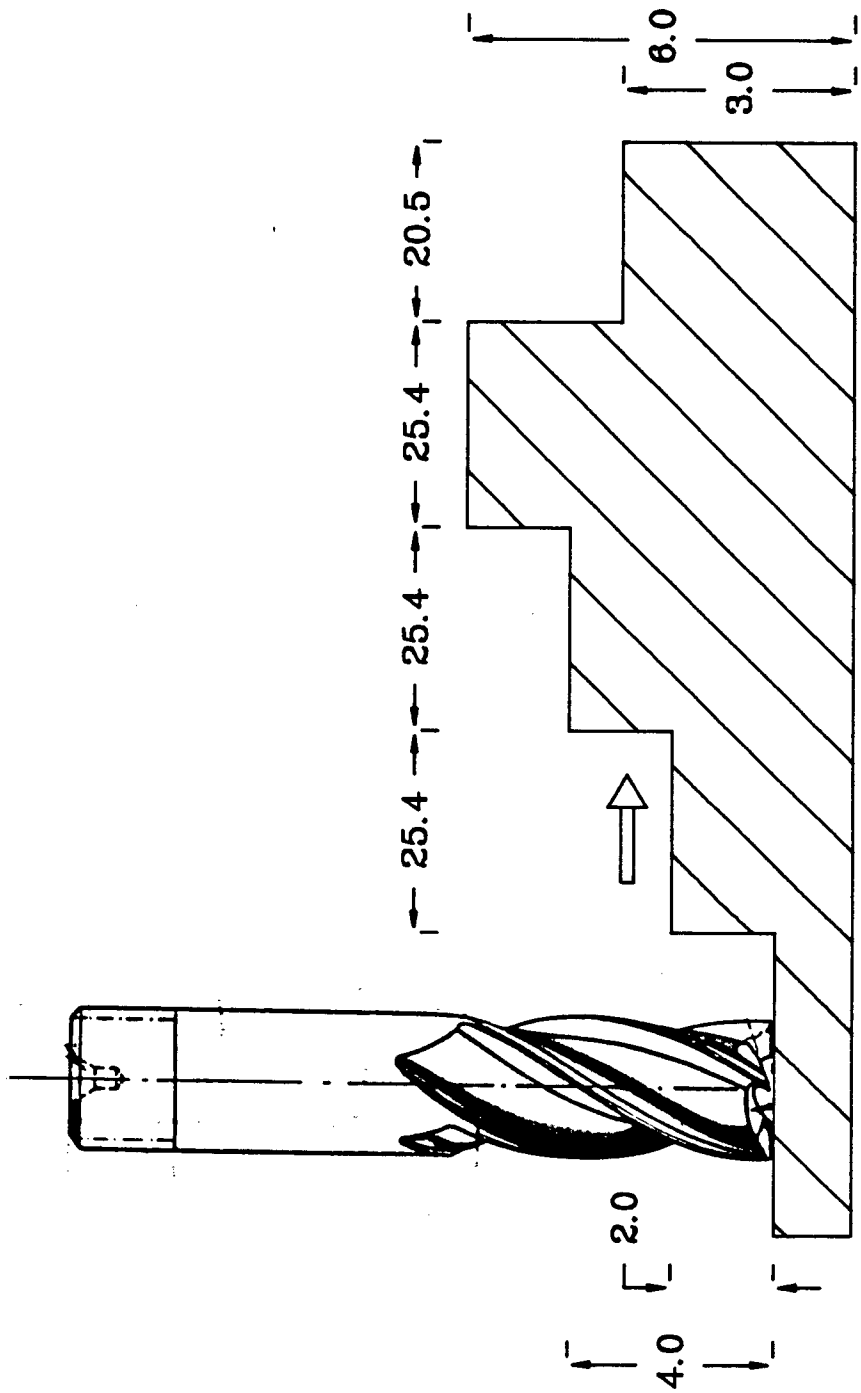

Figure 4.19: Cutting Profile

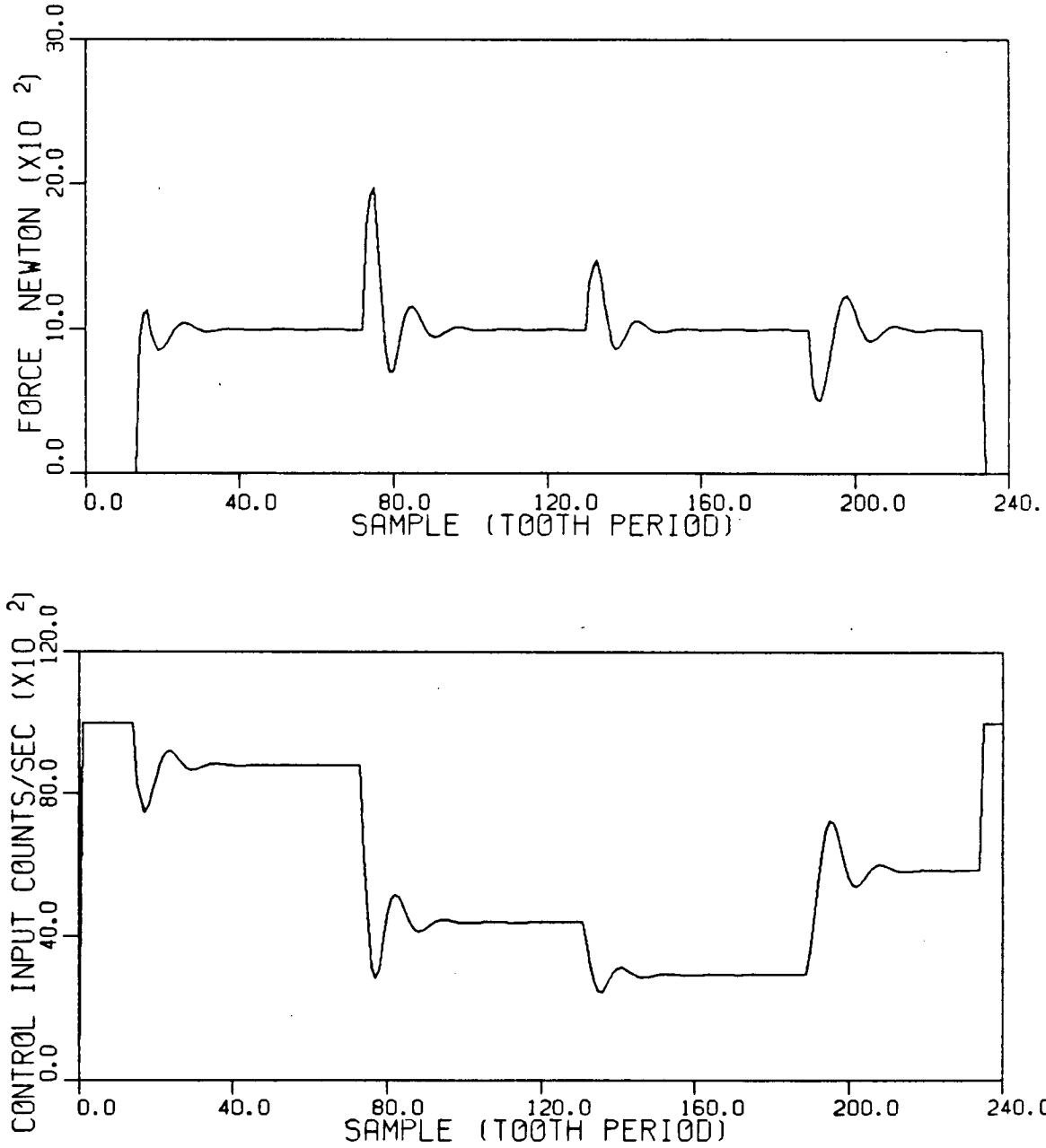

Figure 4.20: PID Controller Simulation
a) Cutting Force Response
b) Control Input

predicted and measured output ,i.e. the predicted error, is greater than 5 percent of the reference force, the covariance matrix is reset to a large value (1000) and forgetting factor decreased to 0.995. This will help the identification method converge faster [41,6,39]. Note that under "steady state conditions" or when there are no changes in the process, the value of the covariance matrix goes to zero and it is commonly said that the estimator "sleeps". When there is a sudden change in axial depth of cut, identification cannot work well and the controller becomes oscillatory. This oscillation although eventually will help the estimator to converge but it is undesirable. Resetting is done infrequently by checking a counter because resetting too frequently is also harmful. A bound is imposed on the pole of the cutting process. The reason is that after every change in the process, it is possible that the few initial estimated parameters are wrong. The pole of the process is limited to the unit circle in these cases.

The same simulation was repeated but this time in the presence of run-out which is simulated by imposing an alternating 10% depth of cut variation at each tooth period. This is important because although run-out was always minimized, by adjusting the inserts on the tool, it still existed. This phenomenon, which acts like a deterministic noise on the system, has a detoriating effect on the controller during cutting[39]. Figure 4.21 shows the cutting force and control input for a PID controller in the presence of run-out. It is clear that the system is responding well.

In a series of tests the PID controller was implemented on the CNC machine. The work-piece material was aluminum 7075T651. Straight slotting was performed to test the algorithm. The spindle speed was 775 RPM and cutter diameter was 2.54 mm. The reference force was 1000 Newtons. To eliminate the measured high frequency noise a low pass filter with break frequency of 100 Hz was used. Force signals in X and Y direction and the tacho signal were passed through the filter. To prevent unexpected control inputs to the machine , which is most likely during the passing steps, the

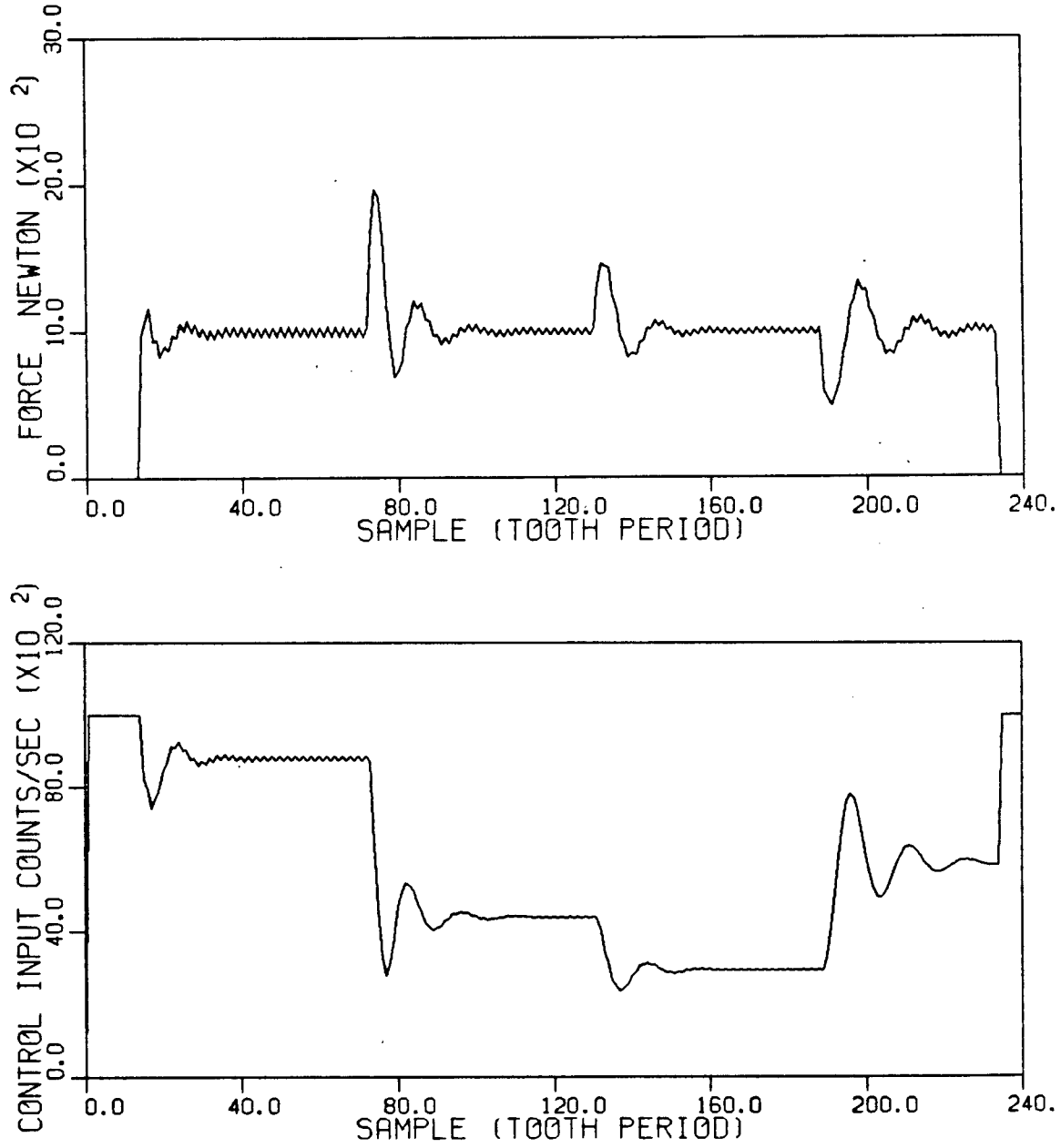

Figure 4.21: PID Controller Simulation with Run-out
a) Cutting Force Response
b) Control Input

controller was bounded to a maximum value. On the other hand to prevent the table from moving backward a lower bound was also imposed on the controller. Before and after machining , or during air cutting, the controller was set to move the table at a constant velocity. The measured peak force is shown in figure 4.22. The peak force is tracking the reference force although there are "spikes" in the force signal during the changes in axial depth of cuts. The control input, the measured feed-rate signal from the tacho and estimated parameters are shown in Figs. 4.23, 4.24 and 4.25 respectively. The control input response is the mirror image of the cutting profile.

One draw-back of this controller is that a thorough knowledge of the servo and cutting process is necessary to implement the pole-zero cancelations. Otherwise the characteristic equation may have higher orders with possible pole or poles outside the unit circle.

The source code for the PID algorithm is given in Appendix A . Supplementary softwares such as data acquisition system, setup and saving data software are given in appendix D .

4.5 Adaptive Pole-Placement Control Design

4.5.1 Introduction

Self tuning controllers with the controller design based on Pole-Placement were first proposed by Wellstead et al.[36] and Astrom and Wittenmark[37]. Here again the model parameters are first identified and then the controller is designed assuming that these parameter estimates are correct. The design procedure explained here is based on the input-output model. The main idea is to choose a linear controller structure, as is shown in figure 4.26, and change its parameters such that the closed loop system has the desired specifications. One advantage of a Pole-Placement controller is that

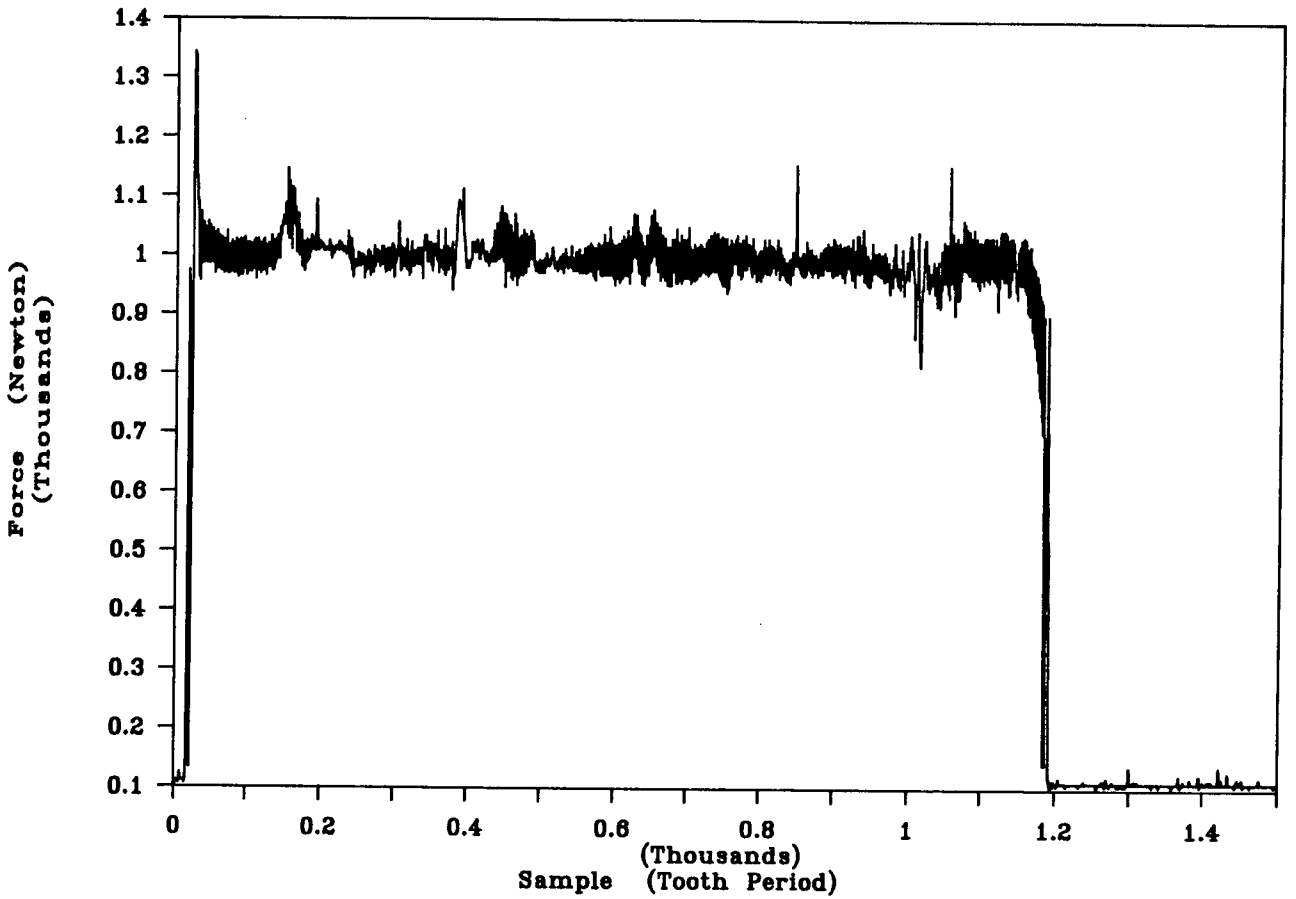

Figure 4.22: Adaptive PID Controller Machining Test, Peak Force Measurement

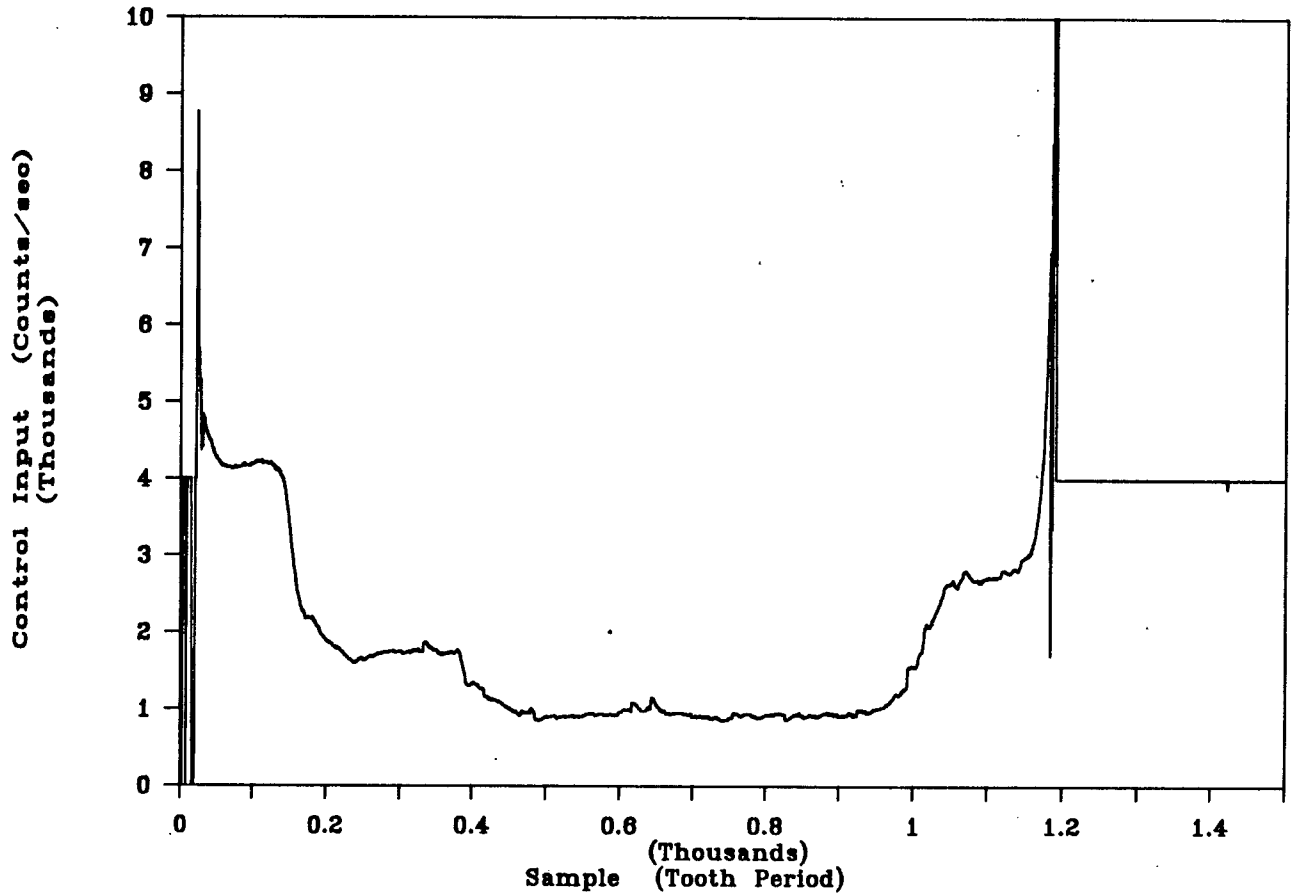

Figure 4.23: Adaptive PID Controller Machining Test, Control Input

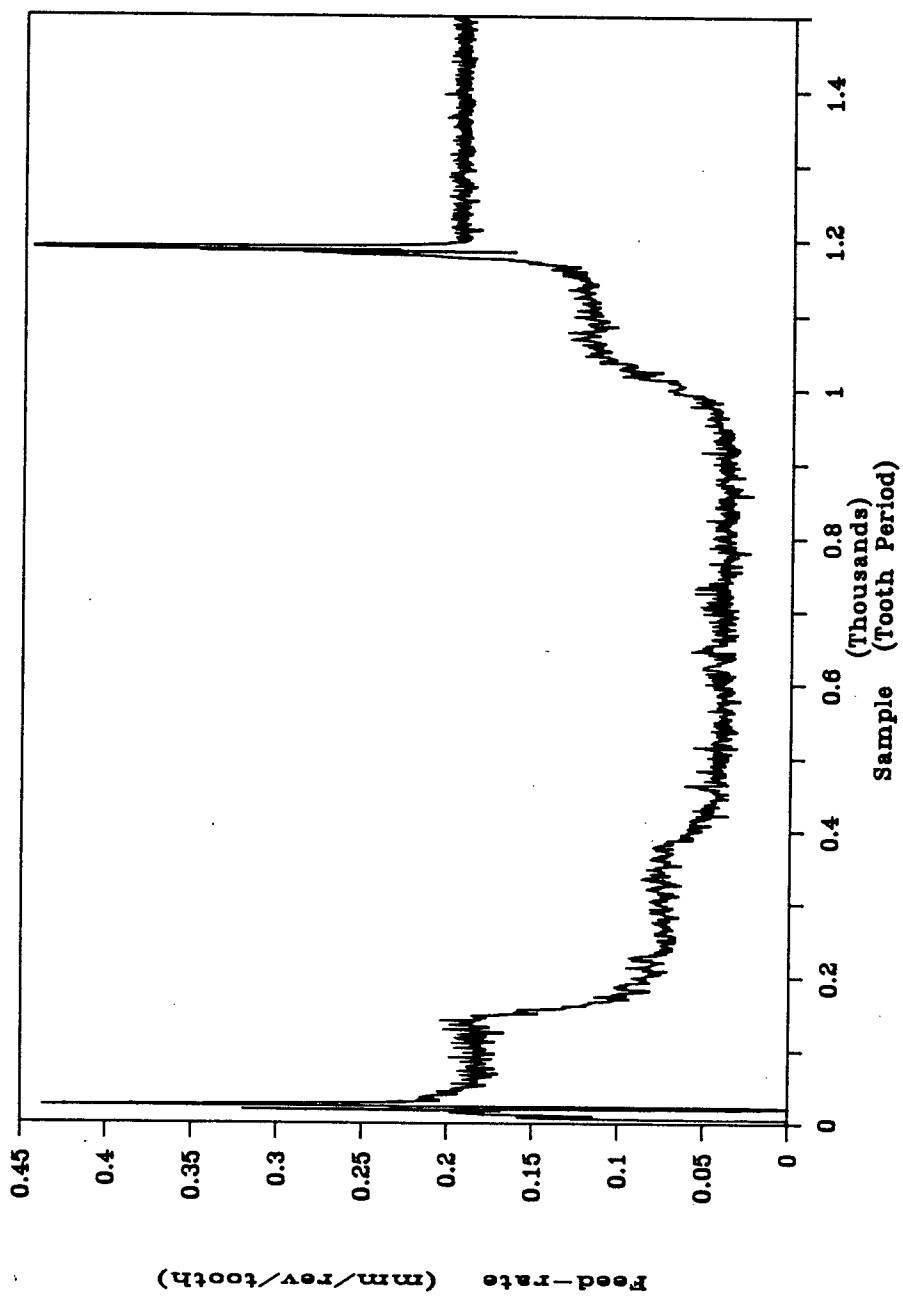

Figure 4.24: Adaptive PID Controller Machining Test, Feed-rate Measurement

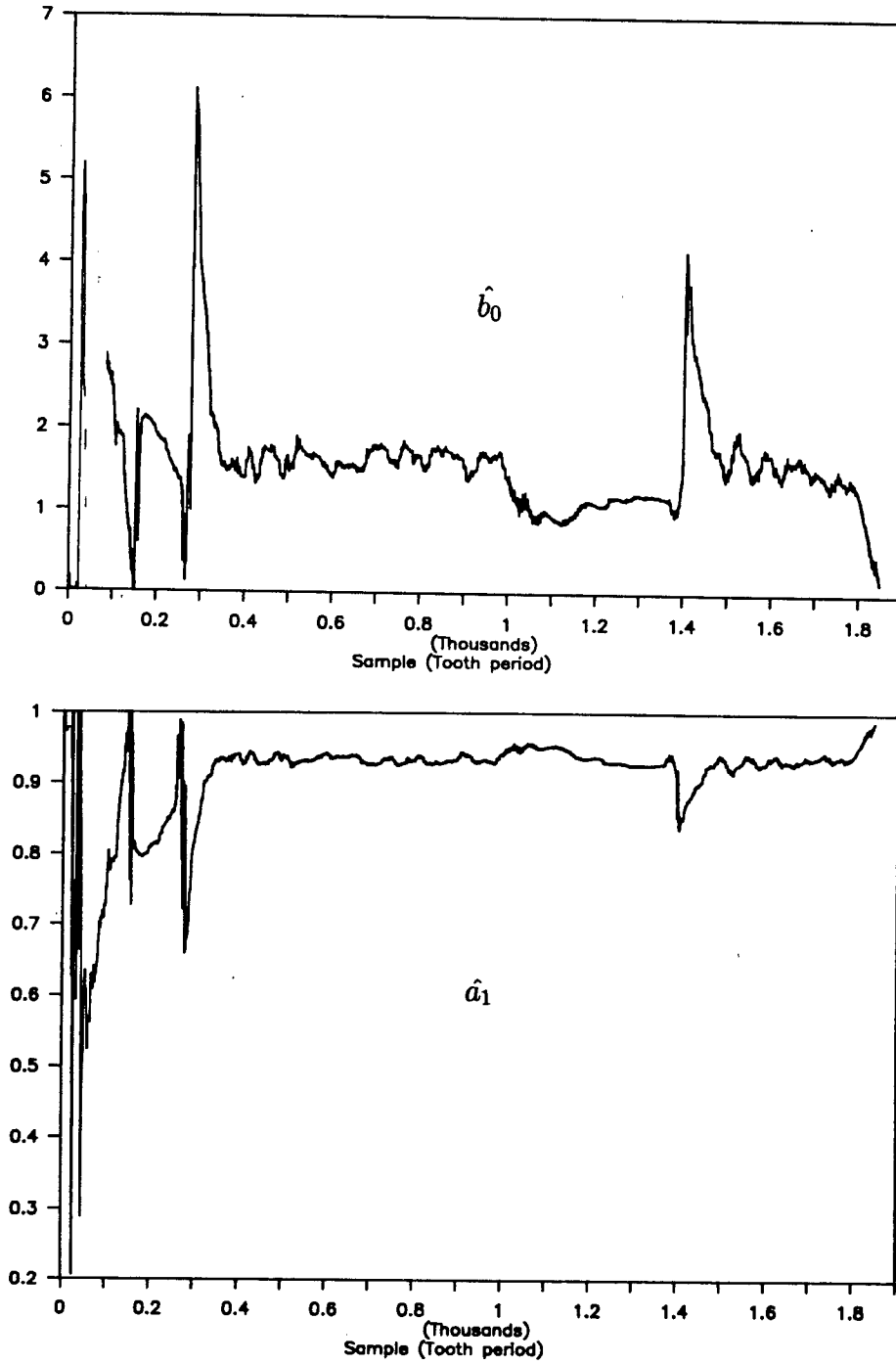

Figure 4.25: Adaptive PID Controller Machining Test, Estimated Parameters

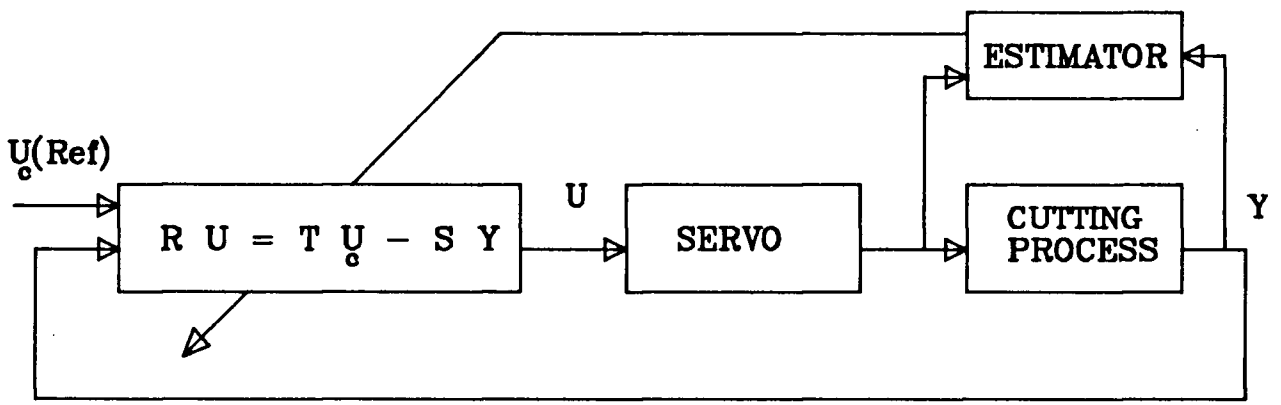

Figure 4.25: Adaptive Pole-Placement Control Loop

by appropriate selection of closed loop poles the stability of the system could be safe-guarded. On the other hand by manipulating the zeros of the closed loop the transient response is improved.

4.5.2 Model Considerations

It is desired to design an adaptive controller which can keep the applied cutting force on the shank of the cutter at a safe constant level. The controller must be stable, robust to changes in the cutting conditions and have an acceptable transient response. The process to be controlled has two cascaded dynamics. The first being the time invariant feed drive control dynamics (see Eq. 4.65), and the second is the time varying cutting process whose parameters change depending on the changes in the work-piece geometry during machining. The combined feed drive control and machining process dynamics are referred to as the *plant*. The plant's transfer function is given as:

$$\frac{y(t)}{u(t)} = \frac{B(q^{-1})}{A(q^{-1})} \tag{4.89}$$

where $y(t)$ and $u(t)$ are the force and control input signal respectively and q is a forward shift time operator. It is assumed that polynomials A and B are relatively prime and $\deg A \geq \deg B$, i.e., the system is causal. The polynomial A is assumed to be monic, i.e., the first coefficient is unity. Substitution of the feed drive control and process parameters into the equation 4.89 gives the plant's transfer function which has all the above requirements.

$$\frac{y(t)}{u(t)} = \frac{k_1(z+\alpha)}{z(z^2 + A_1 z + A_2)} \tag{4.90}$$

where

$$\begin{aligned} k_1 &= K\,\hat{b}_0\,0.9518 \\ \alpha &= 0.0020946 \end{aligned}$$

$$A_1 = -\hat{a}_1 - 0.04614$$
$$A_2 = +0.04614\ \hat{a}_1$$

The cutting process parameters $\hat{a}_1$ and $\hat{b}_0$, are time varying, and are estimated at each tooth period using the RLS identification method.

The aim of the adaptive controller based on Pole-Placement design is that the closed loop transfer function between the reference force and the actual force obey the following desired model dynamics

$$\frac{y(t)}{u_c(t)} = \frac{B_m(q)}{A_m(q)} \tag{4.91}$$

For the stability of the system, the roots of the characteristic equation, A_m, are selected within the unit circle. Furthermore, A_m is designed to satisfy transient response characteristics of the controller. A second order dynamics is selected to represent the desired response

$$A_m(z) = z^2 - 2\ e^{-\zeta\ \omega_n T}\ cos(\omega_n\sqrt{1-\zeta^2}T)\ z + e^{-2\zeta\omega_n T} \tag{4.92}$$

which corresponds to a continuous system with a desired damping ratio of ζ and natural frequency of ω_n. T is the sampling time which is equal to one tooth period. An ideal damping ratio of $\zeta = 0.7$ is selected to obtain minimum overshoot during the transients. A rise time of four tooth periods is selected, which corresponds to the natural frequency of 16 rad/sec [38].

$$\omega_n = \frac{2.5}{4T} = 16\ rad/sec. \tag{4.93}$$

The resulting model characteristic equation is found from equation 4.92 as

$$A_m(z) = z^2 - 1.162\ z + 0.4132 \tag{4.94}$$

In order to have a unit closed loop gain, the numerator of the model is selected as

$$B_m(z) = A_m(z = 1) = 0.2512 \tag{4.95}$$

The desired closed loop transfer function of the system is then given by

$$\frac{B_m(z)}{A_m(z)} = \frac{0.2512}{z^2 - 1.162\, z + 0.4132} \tag{4.96}$$

4.5.3 Theory of Pole-placement Design for Machining Process Control

A general form of a self tuning regulator is shown in Fig. 4.26. The polynomials $S(q^{-1})$, $T(q^{-1})$ and $R(q^{-1})$ represent feed-back, feed-forward and error regulators respectively, which have to be determined adaptively at each tooth period. The plant to be controlled was given by Eq. 4.90 which can be rewritten in difference equation form as Eq. 4.89. The closed loop transfer function of the controller is derived from figure 4.26 as :

$$\frac{y(k)}{u_c(k)} = \frac{B\,T}{AR + B\,S} \tag{4.97}$$

where k is the sampling counter. Note that for simplicity the q operator is omitted as an argument of the polynomials. In order for the system to behave like the desired model, the following equivalence must hold.

$$\frac{B\,T}{AR + BS} = \frac{B_m}{A_m} \tag{4.98}$$

The design problem is defined as finding the polynomials R, S and T using the plant and desired model transfer functions.

The relationship between the desired reference force F_r and regulated feeding velocity u can be expressed as

$$R\,u(k) = T\,u_c(k) - S\,y(k) \tag{4.99}$$

It is shown in state space theory that the regulator given by Eq. 4.99 corresponds to a combination of an observer and a state feed-back [27, pp. 148-150]. The observer is designed so that changes in the command signals do not generate errors in the observer. Although the command signal in milling process control is the reference force which is constant, a general case where the reference force may be varied is considered. This results in a factor which cancels in the right hand side of the Eq. 4.98 and can be interpreted as the observer polynomial A_0, which is selected to have its all zeros within the unit circle. It follows that Eq.4.98 can be written as

$$\frac{B\,T}{AR + BS} = \frac{B_m A_0}{A_m A_0} \tag{4.100}$$

In the plant to be controlled, there is one zero which is well within the unit circle, therefore it can be safely canceled by one of the closed loop poles in order to obtain a desired response. The numerator of the plant can be factored as

$$B = B^+ B^- \tag{4.101}$$

where

$$\begin{aligned} B^+ &= z + 0.002046 \\ B^- &= 0.9518\, K\, \hat{b}_0 \end{aligned}$$

Therefore B^- is a constant. In order to cancel B^+ with one of the poles of the closed loop transfer function, as A and B are coprime, following condition must be satisfied by the polynomial R.

$$R = B^+\, R' \tag{4.102}$$

Since B^- is retained, it is not a factor of the characteristic equation of the closed loop system, then it must be a factor of B_m to satisfy the Eq. 4.100.

$$B_m = B^- \, B_m' \tag{4.103}$$

Substituting equations 4.101, 4.102 and 4.103 into Eq. 4.100 gives,

$$\frac{B^- \, T}{AR' + B^- S} = \frac{B^- B_m' A_0}{A_m A_0} \tag{4.104}$$

In order to satisfy Eq. 4.104, the following conditions must hold

$$AR' + B^- S = A_m A_0 \tag{4.105}$$

$$T = B_m' A_0 \tag{4.106}$$

In Eq. 4.105, the polynomials A, A_m and A_0 are known, and polynomials R' and S are to be solved. In order to have a unique solution for the *Diophantine* equation (Eq. 4.105), the following condition must be met [37, page 228]

$$deg \; S \; \leq \; deg \; A \tag{4.107}$$

The degree of polynomial S is chosen to be

$$deg \; S \; = \; deg \; A - 1 \tag{4.108}$$

Furthermore, the following conditions must be satisfied in order for the control law given in Eq. 4.99 to be casual [37, pp. 230-31]

$$deg \; A_0 \; \geq \; 2deg \; A - deg \; A_m - deg \; B^+ - 1 \tag{4.109}$$

$$deg \; R \; = \; deg \; A_0 + deg \; A_m + deg \; B^+ - deg \; A \tag{4.110}$$

4.5.4 Application of Pole-Placement Design for Milling Process Control

The Pole-Placement design method recapitulated in the previous section has been applied to control the milling process represented by the plant given in Eq. 4.90. It is desired that the milling process behave according to the model selected in Eq. 4.96.

The conditions listed in equations 4.108 and 4.109 are applied to the plant and the model to find the order of the polynomials.

$$\begin{aligned} deg\ A_0 &= 2 \\ deg\ S &= 2 \\ deg\ R &= 2 \end{aligned}$$

As a rule of thumb the observer dynamics should be faster than the desired closed loop response [41, pp. 502-503], therefore a deadbeat observer is selected as

$$A_0 = z^2 \tag{4.111}$$

Since a factor of the polynomial R is canceled by the plant zero B^+ (Eq.4.102), the polynomial R' can be expressed as a first order polynomial

$$R' = z + r_1 \tag{4.112}$$

The feed-back regulator S is expressed as :

$$S(z) = s_0\, z^2 + s_1\, z + s_2 \tag{4.113}$$

Substitution of polynomials into the Eq. 4.105 gives the *Diophantine* Equation

$$\begin{aligned} z\,(\,z\,-\,0.0461992\,)\,(z\,-\,\hat{a}_1)\,(z\,+\,r_1)\,+ \\ (\,s_0\,z^2\,+\,s_1\,z\,+\,s_2)\,(K)\,=z^2\,(z^2\,+\,-1.162z\,+\,0.4132) \end{aligned} \tag{4.114}$$

this results in

$$r_1 = -1.1158 + \hat{a}_1 \tag{4.115}$$

$$s_0 = \frac{0.4132 + r_1\,\hat{a}_1 + 0.0461992\,(r_1 - a_1)}{k_1} \tag{4.116}$$

$$s_1 = \frac{-0.0461992\,(r_1\,\hat{a}_1)}{k_1} \tag{4.117}$$

$$s_2 = 0. \tag{4.118}$$

r_1 and s_0 and s_1 are calculated from the above equations at each sampling time and then polynomials S and R are calculated as follows

$$S = z(\,s_0\,z + s_1) \tag{4.119}$$

and from Eq. 4.102

$$R = B^+\,R' = (z + 0.0020946)\,(z + r_1) \tag{4.120}$$

Polynomial T is calculated from Eq.4.106

$$T = B'_m\,A_0 = z^2\,\frac{0.2639}{K\,\hat{b}_0} \tag{4.121}$$

The machining process parameters $\hat{b}_0$ and $\hat{a}_1$ are identified recursively at each control interval (i.e. tooth period) to adaptively determine the polynomial parameters given in equations 4.119,4.120 and 4.121.

These polynomials satisfy the requirements of causality

$$deg\,R \leq deg\,T$$

$$deg\,R \leq deg\,S$$

The control input signal is given as

$$R(q^{-1})\,u(k) = T(q^{-1})\,u_c(k) - S(q^{-1})\,y(k) \tag{4.122}$$

Table 4.3 presents a psuedo code for Pole-Placement control algorithm.

Table 4.2: Psuedo Code for PID Control Algorithm

1) Measure the peak resultant force and tacho signal

2) Identify the parameters of the model based on procedure presented in Table 4.1.

3) From Eqs. 4.88, 4.71 and 4.72 find the gains K_p, K_I and K_d.

4) Calculate error signal by subtracting measured cutting force from reference input signal.

5)Calculate control input using Eq. 4.68 with known gains and error signal as input.

6) Wait until next sampling time, go to step one and repeat.

Table 4.3: Psuedo Code for Pole-Placement Control Algorithm

1) Measure peak resultant force and tacho signal.

2) Identify the parameters of the model based on procedure described in Table 4.1.

3) Calculate polynomials R, S and T according to Eqs. 4.120 , 4.119 and 4.121.

4) Calculate control input signal from Eq. 4.122.

5) Wait until next sampling time, go to step one and repeat.

4.5.5 Simulation and Experimental Results

The simulation results for Pole-Placement controller is shown in figure 4.27. The same cutting profile and reference force value was used as in case of PID controller. The algorithm was also simulated in the presence of run-out. The cutting force response and control input are shown in Fig. 4.28. As it is shown in simulations the pole-placement controller can follow the reference force even with sudden changes in axial depth of cut. The response is similar to that of the PID controller because the characteristic equations for both designs are almost the same.

The Pole-Placement algorithm was run on the CNC machine under the same conditions as PID controller to set a base for comparison. Parameter Identification has the same procedure as the case of the PID to prevent drifting of parameters. During steady state cutting, if the predicted error was less than a minimum value (one percent of the reference force) then the previous control input value is sent to the servo. The measured cutting force is shown in Fig. 4.29. The output is tracking the reference signal well. Note that the height of the jumps in machining tests are generally lower than the simulations. The reason is that during cutting, the cutter gradually penetrates the flat face of the work-piece, but in simulations the changes are treated once. Control input signal, measured tacho signal and estimated parameters are shown in figures 4.30, 4.31 and 4.32 respectively. The parameters converge slowly to their true values. Change from axial depth of cut of 6 to 3 mm shows the same response in simulation and experiment. Note that after a 'burst-like' response the parameters converge fast to their final values and the controller works satisfactory.

One drawback of the pole-placement technique is that the structure of the plant should be known before hand. If the model and the process does not match then it is possible for the controller to become unstable.

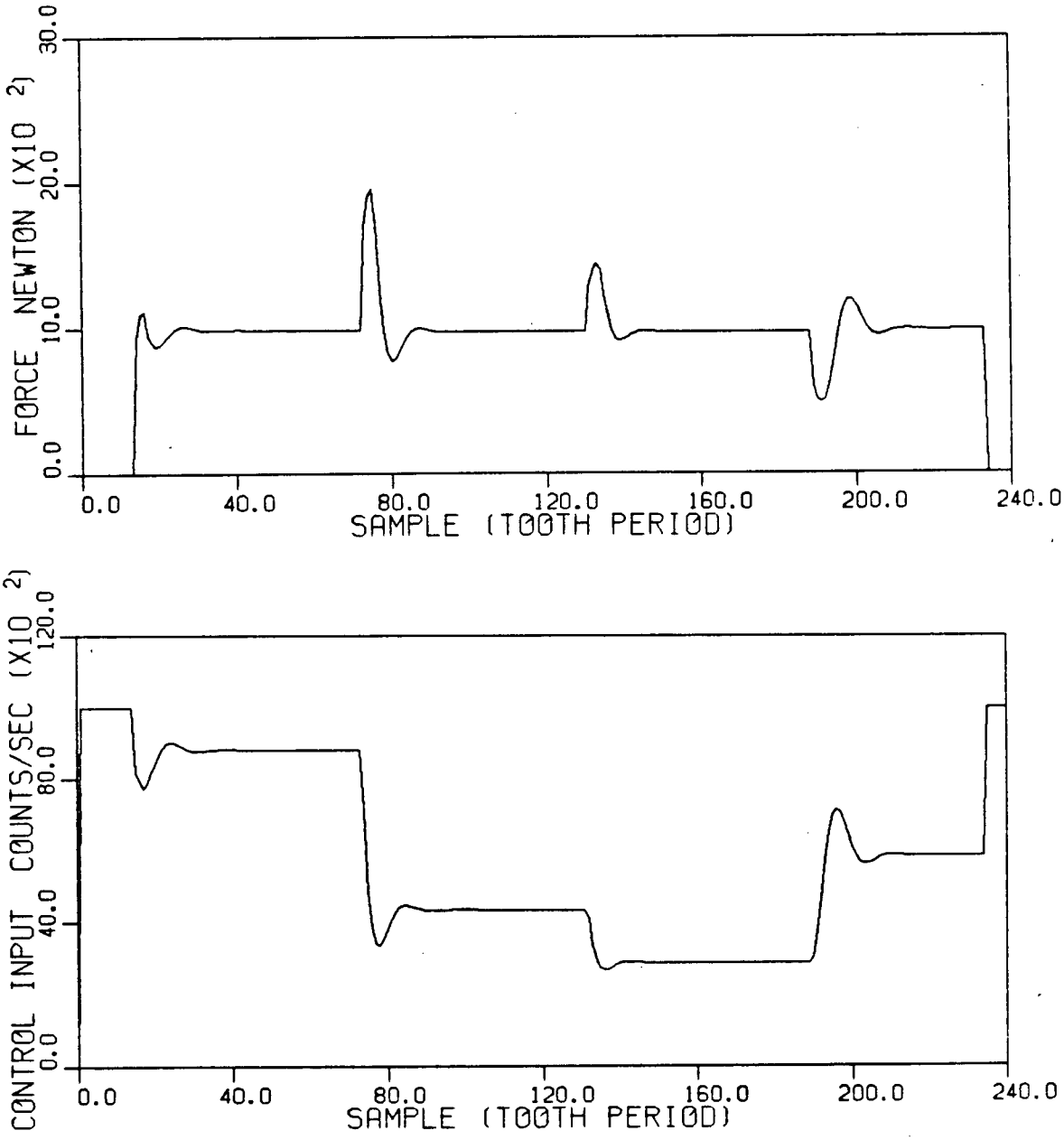

Figure 4.26: Pole-Placement Controller Simulation
a) Cutting Force Response
b)Control Input

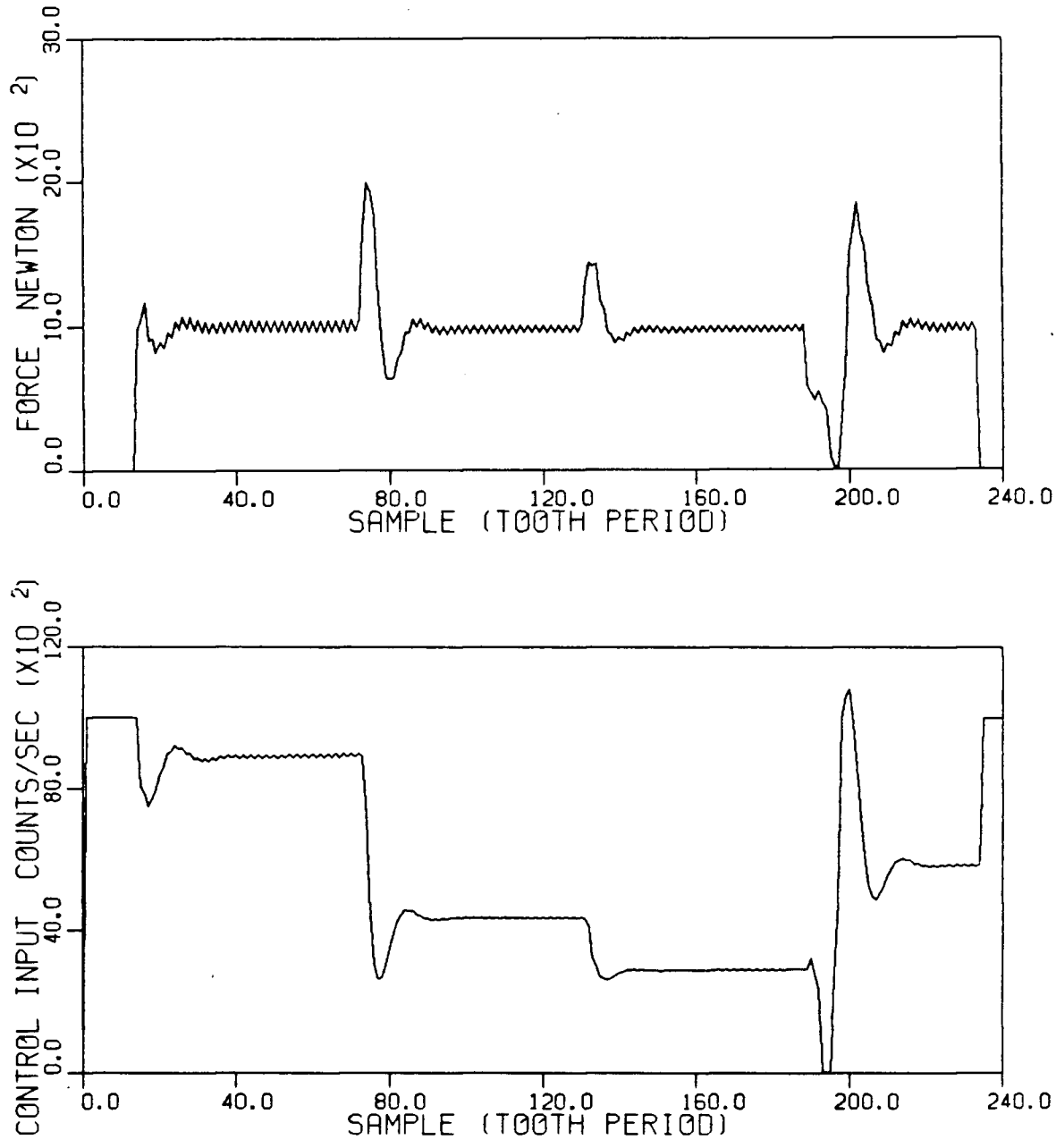

Figure 4.27: Pole-Placement Simulation with Run out
a) Cutting Force Response
b)Control Input

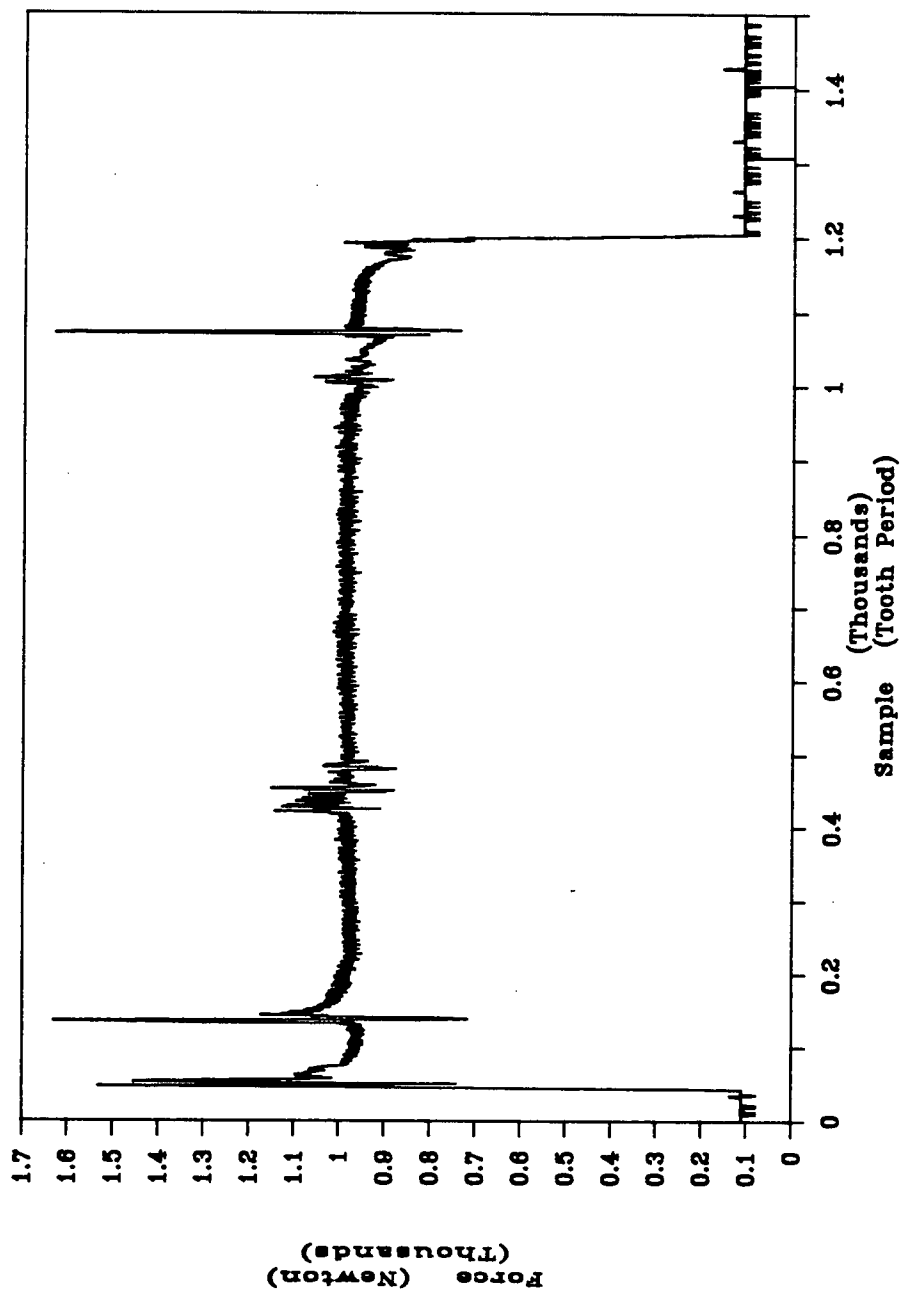

Figure 4.28: Adaptive Pole-Placement Machining Test, Peak Force Measurement

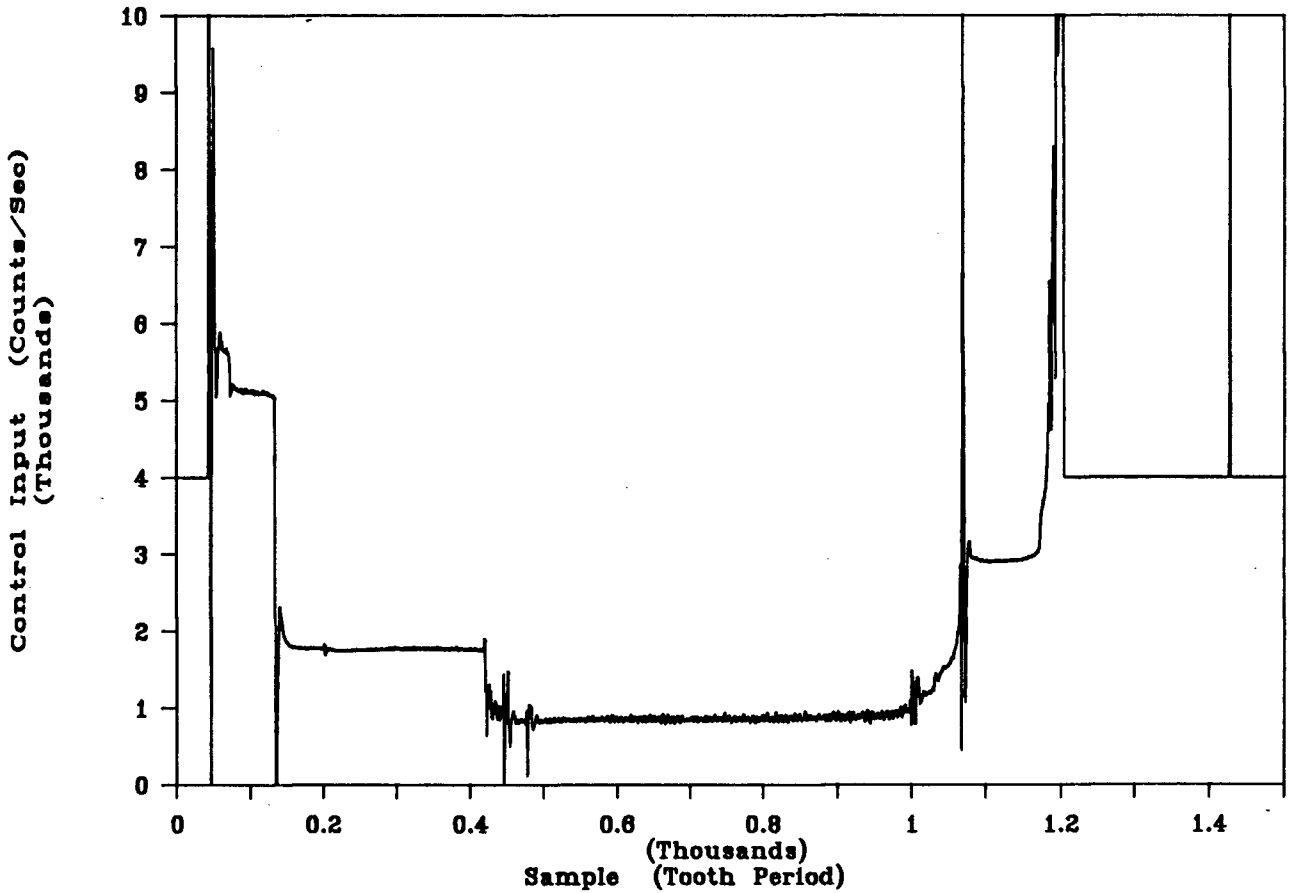

Figure 4.29: Adaptive Pole-Placement Machining Test, Control Input

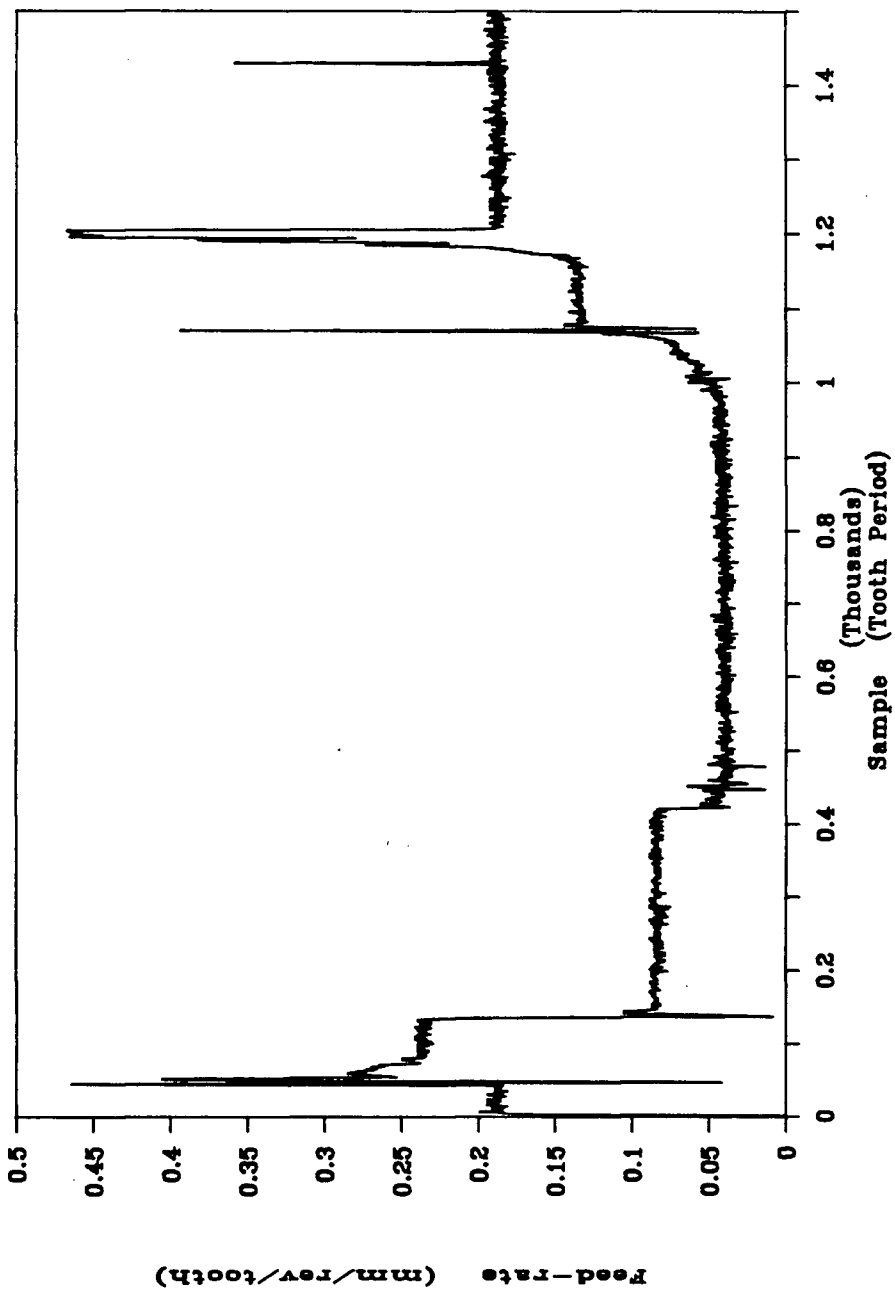

Figure 4.30: Adaptive Pole-Placement Machining Test, Feed-rate Measurement

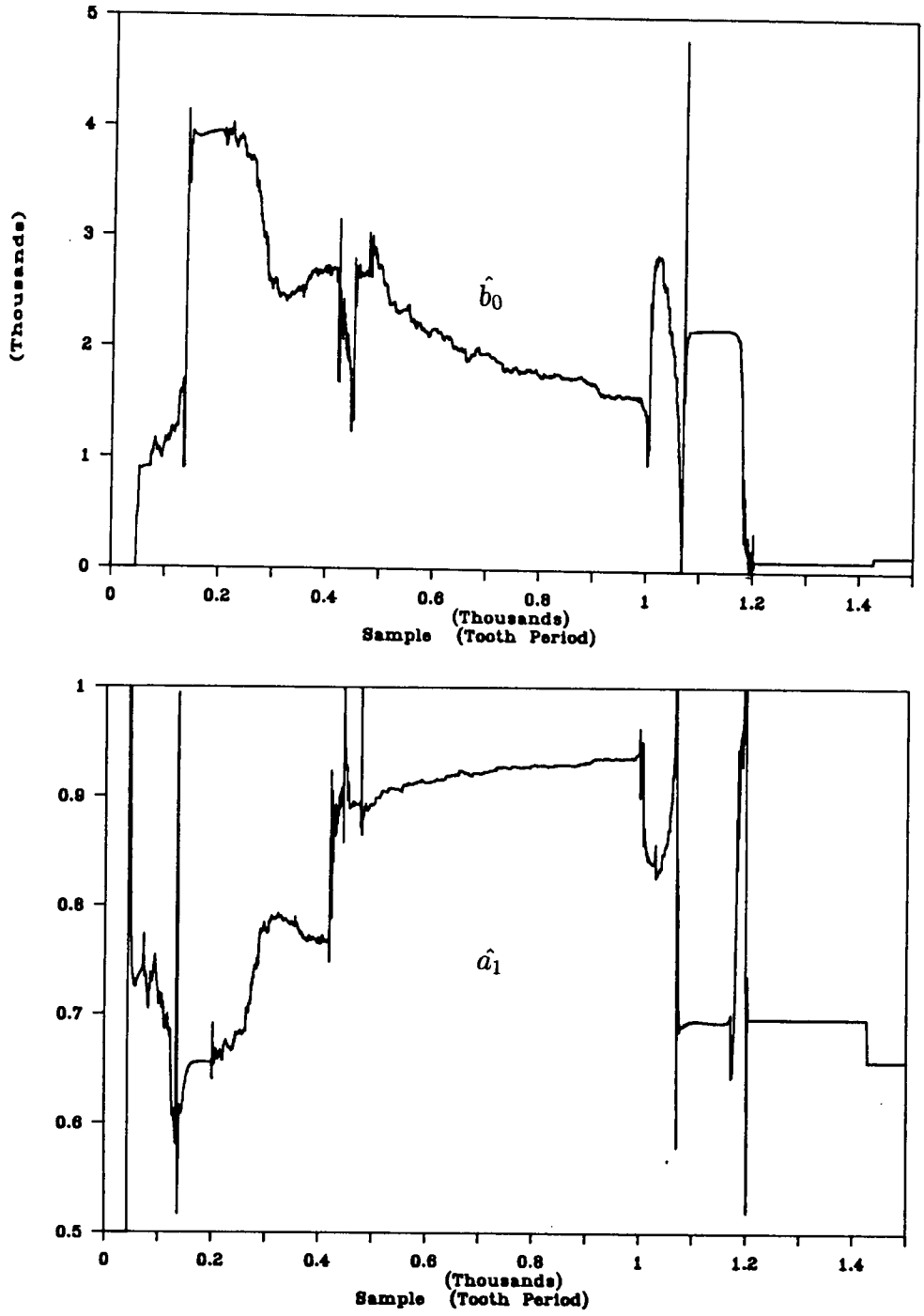

Figure 4.32: Adaptive Pole-Placement Machining Test, Estimated Parameters

Appendix B gives the source code for the pole-placement algorithm.

4.6 Model Reference Adaptive Control Design

4.6.1 Introduction

The first attempt in the design of stable adaptive control systems was in the area of model reference adaptive control systems [6] and was proposed by Whitaker et al.[8] in the late 1950s. Meanwhile, there has been a fair amount of work in this area, especially regarding the stability of the controller. The original MRAC introduced two new ideas[41]. First, the performance of a system was specified by a model; second, the parameters of the controller were adjusted based on the error between the reference model and the system such that the output of the unknown system asymptotically approached that of a given reference model (Fig. 2.2).

MRAC systems were originally derived for servo problems in deterministic continuous-time systems, but the theory has been extended to cover discrete-time systems and systems with stochastic disturbances. Earlier work in this area were based on the state space formulation, but from a practical point of view the adaptive control systems based on input-output signals are the main interest in the design of stable MRAC controllers. In the following section a basic summary of Landau's model reference adaptive control scheme, which is based on independent tracking and regulation, is presented. A more detailed analysis can be found in [21].

4.6.2 Theory of MRAC for Machining Process Control

Model following problem

MRAC is based on *model following* problem. The model following problem is that for a time-invariant plant the controller parameters are adjusted in a manner that the closed

loop transfer function is close to a specific model.

consider a single input single output discrete linear time-invariant plant as

$$A(q^{-1})y(k) = q^{-d}B(q^{-1})u(k) \tag{4.123}$$

where

$$A(q^{-1}) = 1 + a_1q^{-1} + \ldots + a_{nA}q^{-nA} \tag{4.124}$$

$$B(q^{-1}) = b_0 + b_1q^{-1} + \ldots + b_{nB}q^{-nB} \tag{4.125}$$

d is the plant time delay, q^{-1} is the backward shift operator and u(k) and y(k) are the plant input and output, respectively. It is assumed all zeros of the plant are inside the unit circle, which holds true for the milling process and servo. In machining, the required objectives for the controller are that the cutting forces as output follow a reference input and also the system should eliminate the disturbances and remain stable. These conditions could be considered as

1) In tracking the output should satisfy the equation

$$C_1(q^{-1})y(k) = q^{-d}D(q^{-1})u^M(k) \tag{4.126}$$

where $u^M(k)$ is the reference input. $C_1(q^{-1})$ and $D(q^{-1})$ are polynomials which specify an explicit reference model as

$$C_1(q^{-1})y^M(k) = q^{-d}D(q^{-1})u^M(k) \tag{4.127}$$

where $y^M(k)$ is the model output. $C_1(q^{-1})$ should be a stable polynomial.

2) In regulation , $[u^M(k) = 0]$, an initial disturbance $(y(0) \neq 0)$ is eliminated by the dynamics defined by

$$C_2(q^{-1})y(k+d) = 0k \geq 0 \tag{4.128}$$

where $C_2(q^{-1}) = 1 + C_1^2q^{-1} + \ldots + C_{nc2}^2q^{-nc2}$ is a stable polynomial.

The plant-model error is defined as

$$\varepsilon(k) = y(k) - y^M(k) \tag{4.129}$$

The control objectives in Equations 4.126 and 4.128 are satisfied if the following equation holds

$$C_2(q^{-1})\varepsilon(k+d) = 0 \tag{4.130}$$

The controller structure could be chosen in general form same as the Pole-Placement design

$$S'(q^{-1})u(k) = T'(q^{-1})u^M(k) - R'(q^{-1})y(k) \tag{4.131}$$

One way to choose polynomials $S'(q^{-1})$,$T'(q^{-1})$ and $R'(q^{-1})$ is

$$S'(q^{-1}) = B(q^{-1})S(q^{-1}) \tag{4.132}$$

$$T'(q^{-1}) = C_2(q^{-1})\frac{D(q^{-1})}{C_1(q^{-1})} \tag{4.133}$$

$$R'(q^{-1}) = R(q^{-1}) \tag{4.134}$$

where

$$S(q^{-1}) = 1 + s_1q^{-1} + s_2q^{-2} + \ldots + s_{nS}q^{-nS} \tag{4.135}$$

$$R(q^{-1}) = r_0 + r_1q^{-1} + r_2q^{-2} + \ldots + r_{nR}q^{-nR} \tag{4.136}$$

therefore the controller becomes

$$BSu(k) = C_2\frac{D}{C_1}u^M(k) - Ry(k) \tag{4.137}$$

Note that for simplicity the arguments of the polynomials are omitted.

With the help of equations 4.123 and 4.126 Eq. 4.137 can be rewritten as

$$BS\frac{A}{B}y(k+d) = C_2\frac{D}{C_1}\frac{C_1}{D}y(k+d) - Ry(k) \tag{4.138}$$

or

$$AS + Rq^{-d} = C_2 \tag{4.139}$$

This equation has a unique solution provided that

$$n_S = d - 1$$

$$n_R = max(n_A - 1, n_{c2} - 2) \tag{4.140}$$

With the help of Eq. 4.139 it is straight forward to show that the required control objective, Eq.4.142 ,holds

$$C_2\varepsilon(k+d) = C_2y(k+d) - C_2y^M(k+d) \tag{4.141}$$

using Eqs. 4.141, 4.139 and 4.127

$$C_2\varepsilon(k+d) = (AS + Rq^{-d})y(k+d) - C_2\frac{D}{C_1}u^M(k) \tag{4.142}$$

or

$$C_2\varepsilon(k+d) = BSu(k) + Ry(k) - C_2\frac{D}{C_1}u^M(k) \tag{4.143}$$

Referring to Eq.4.137 the right hand side of the above equation is zero, therefore Eq. 4.130 is satisfied.

Polynomials in the controller, Eq.4.137, could be expanded and rewritten as

$$b_0u(k) + \theta_0^T\phi_0(k) = C_2y^M(k+d) - Ry(k) \tag{4.144}$$

where

$$\phi_0^T(k) = [u(k-1), \ldots, u(k-d-n_B+1), y(k), \ldots, y(k-n_R)] \tag{4.145}$$

$$\theta_0^T = [b_0s_1 + b_1, b_0s_2 + b_1s_1 + b_2, \ldots, b_{nB}s_{d-1}, r_0, \ldots, r_{nR}] \tag{4.146}$$

Finally control input can be derived from Eq. 4.144 as

$$u(k) = \frac{1}{b_0}[C_2(q^{-1})y^M(k+d) - \theta_0^T\phi_0(k)] \tag{4.147}$$

Figure 4.33 shows the block diagram of the linear model following control scheme.

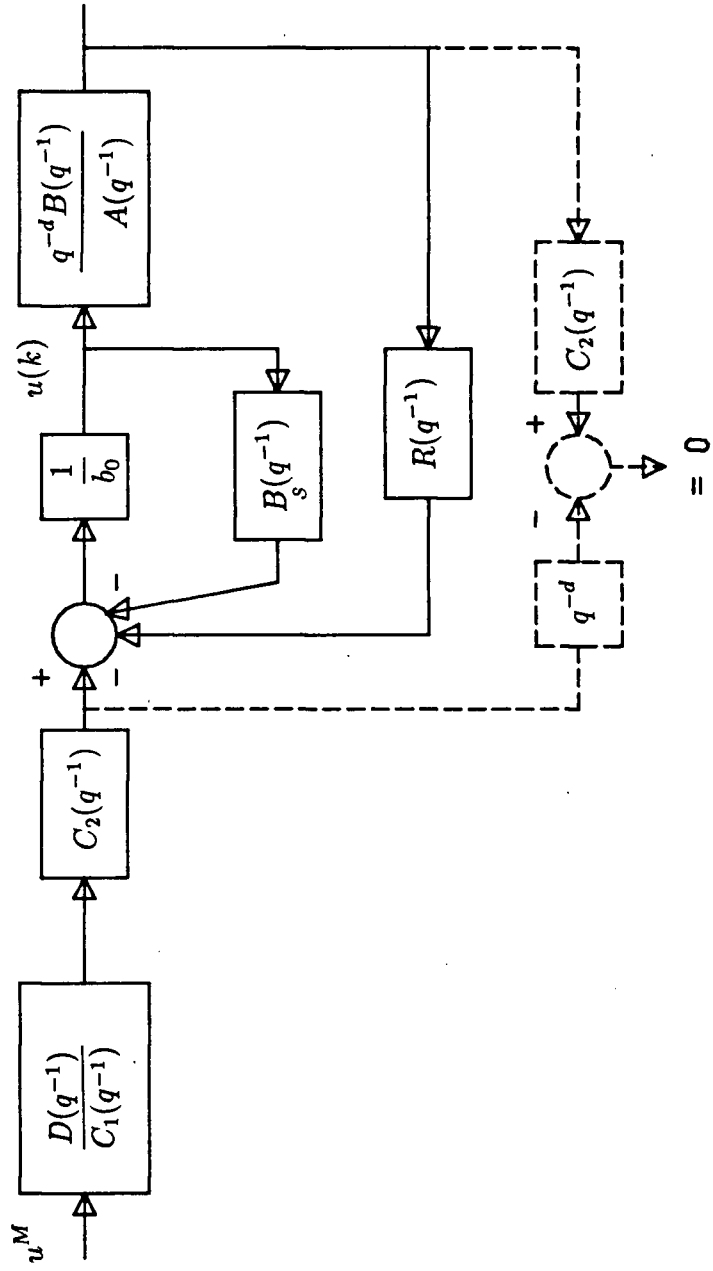

Figure 4.31: Tracking and Regulation Control Block for a Known Plant

MRAC Design

When the plant parameters are unknown, they are replaced by their estimates from the adaptation mechanism. Therefore the control law in the adaptive case is given by

$$u(k) = \frac{1}{\hat{b}_0(k)}[C_2(q^{-1})y^M(k+d) - \hat{\theta}_0^T(k)\phi_0(k)] \tag{4.148}$$

or equivalently

$$\hat{\theta}^T(k)\phi(k) = C_2(q^1)y^M(k+d) \tag{4.149}$$

where

$$\hat{\theta}^T(k) = [\hat{b}_0(k); \hat{\theta}_0^T(k)] \tag{4.150}$$

$$\phi^T(k) = [u(k); \phi_0^T] \tag{4.151}$$

Introducing Eqs. 4.149 and 4.139 into Eq. 4.142 gives

$$C_2(q^{-1})\varepsilon(k+d) = [\theta - \hat{\theta}(k)]\phi(k) \tag{4.152}$$

The design objective is to find an adaptation mechanism which ensures that the plant-model error converges to zero and the input and output of the plant remain bounded.

$$\lim_{k\to\infty} \varepsilon(k) = 0$$

$$\| \phi(k) \| \leq M < \infty, \forall k \tag{4.153}$$

Because the plant parameters are unknown Eq. 4.152 cannot be solved directly, however, it can be approached as follows.

Define the filtered plant-model error as

$$\varepsilon^f(k) = C_2(q^{-1}\varepsilon(k) = [\theta - \hat{\theta}(k-d)]^T\phi(k-d) \tag{4.154}$$

Define the auxiliary error as

$$\bar{\varepsilon}(k) = [\hat{\theta}(k-d) - \hat{\theta}(k)]^T\phi(k-d) \tag{4.155}$$

The summation of the above errors gives the augmented error

$$\varepsilon^*(k) = \varepsilon^f(k) + \bar{\varepsilon}(k) \tag{4.156}$$

It can be proved [21] that if

$$\lim_{k\to\infty} \varepsilon^*(k) = 0 \tag{4.157}$$

then the objectives of equations 4.153 are satisfied. An adaptation mechanism which guarantees the conditions of Eq. 4.157 is given as

$$\hat{\theta}(k) = \hat{\theta}(k-1) + P_k\phi(k-d)\varepsilon^*(k) \tag{4.158}$$

$$P_{k+1} = \frac{1}{\lambda_1(k)}[P_k - \frac{P_k\phi(k-d)\phi^T(k-d)P_k}{\lambda_1(k)/\lambda_2(k) + \phi^T(k-d)P_k\phi(k-d)}] \tag{4.159}$$

where $0 < \lambda_1(k) \leq 1$, $0 \leq \lambda_2(k) < 2$ and $P_0 > 0$

To make the algorithm implementable , $\varepsilon^*(k)$, augmented error , should be expressed in terms of parameters estimated up to $k-1$. Using Eqs. 4.156, 4.155, 4.154 and 4.158

$$\begin{aligned} \varepsilon^*(k) &= \varepsilon^f(k) + \bar{\varepsilon}(k) \\ &= C_2(q^{-1})\varepsilon(k) + \hat{\theta}^T(k-d)\phi(k-d) - \hat{\theta}^T(k)\phi(k-d) \end{aligned} \tag{4.160}$$

or

$$\begin{aligned} \varepsilon^*(k) &= C_2(q^{-1})y(k) - \hat{\theta}^T(k)\phi(k-d) \\ &= C_2(q^{-1})y(k) - \hat{\theta}^T(k-1)\phi(k-d) - \phi^T(k-d)P_k\phi(k-d)\varepsilon^*(k) \end{aligned} \tag{4.161}$$

finally

$$\varepsilon^*(k) = \frac{C_2(q^{-1})y(k) - \hat{\theta}^T(k-1)\phi(k-d)}{1 + \phi^T(k-d)P_k\phi(k-d)} \tag{4.162}$$

Figure 4.34 shows the adaptive control scheme corresponding to this design.

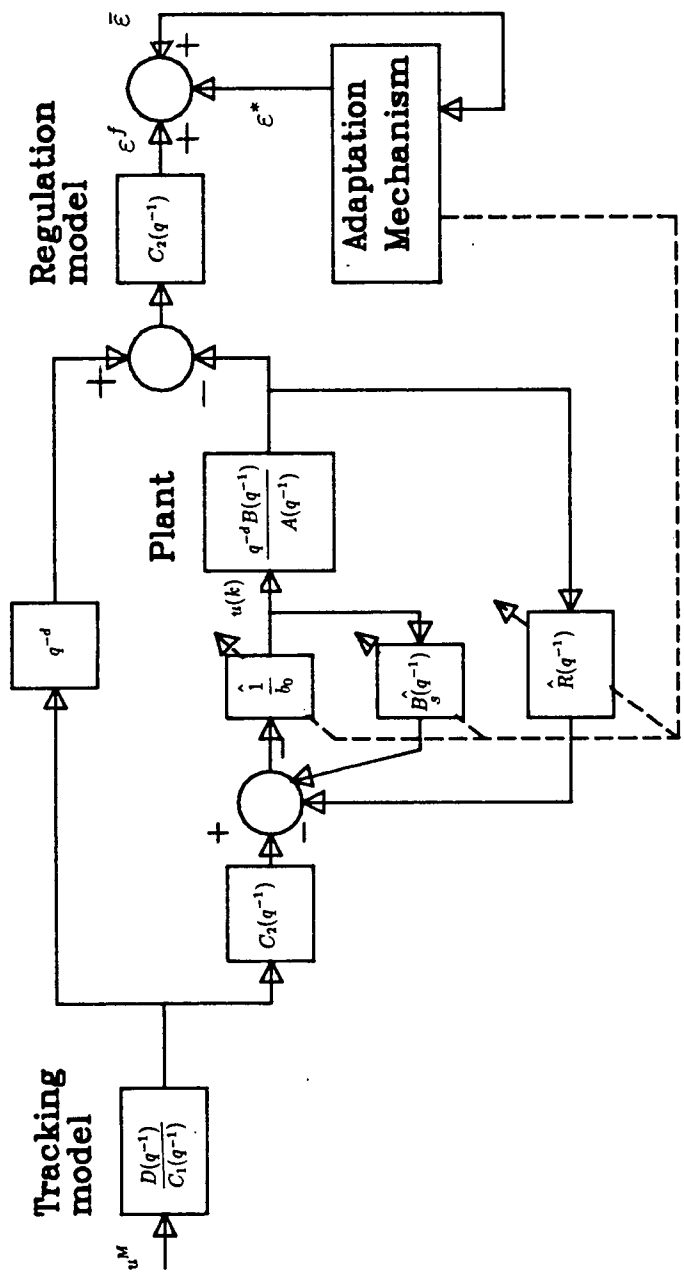

Figure 4.32: Tracking and Regulation Adaptive Control Block

4.6.3 Application of MRAC Design for Milling Process Control

The MRAC design method has been applied to control the milling process. The plant is the combination of the servo feeding velocity and cutting process transfer functions.

$$\frac{y(k)}{u(k)} = \frac{K(0.99578086)b_0 z^{-2}(1 + 0.0020946z^{-1})}{(1 - 0.0461992z^{-1})(1 - a_1 z^{-1})} \tag{4.163}$$

Comparing with the Eq. 4.123 one gets

$$n_A = 2, n_B = 1, d = 2 \tag{4.164}$$

The choice of regulation dynamics, C_2, is by designer and through simulations a second order polynomial is selected as

$$C_2(z^{-1}) = 1 + C_1^2 z^{-1} + C_2^2 z^{-2} = 1 - 0.44z^{-1} + 0.016z^{-2} \tag{4.165}$$

consequently the order of polynomials $S(q^{-1})$ and $R(q^{-1})$ are given by Eq.4.140

$$n_S = 1, n_R = 1 \tag{4.166}$$

The reference model has been chosen with the same structure as the plant. The parameters of the cutting process are calculated based on average depth of cut of 5 mm and the resulting transfer function is multiplied by the time-invariant transfer function of the servo. This model is multiplied by a gain of 2.235 to obtain a dc gain of one for the model.

$$\frac{y^M(k)}{u^M(k)} = \frac{D}{C_1} = \frac{2.235z^{-2}(0.3 + 0.00062838z^{-1})}{(1 - 0.0461992z^{-1})(1 - 0.3z^{-1})} \tag{4.167}$$

The observation and parameter vectors are

$$\phi^T(k) = [u(k), u(k-1), u(k-2), y(k), y(k-1)] \tag{4.168}$$

$$\hat{\theta}^T(k) = \left[\hat{b}_0(k), \hat{b}_0(k)\hat{s}_1(k) + \hat{b}_1, \hat{b}_1\hat{s}_1, \hat{r}_0, \hat{r}_1\right] \tag{4.169}$$

Table 4.4 shows a psuedo code for MRAC algorithm.

Table 4.4: Psuedo Code for MRAC Design

1) Choose the parameters of the regulation dynamics,C_2.

2) Choose the parameters of polynomials D, C_1, model, as Eq. 4.163.

3) Calculate the order of polynomials R and S from Eq. 4.166.

4) Setup observation and parameter vectors as Eqs. 4.168, 4.169.

5) Give initial values to parameter vector and covariance matrix.

6) Measure peak resultant force.

7) Calculate augmented error from equation 4.162.

8) Calculate new parameter vector from Eq. 4.158.

9) Update covariance matrix from Eq. 4.159.

10) Calculate control input signal from Eq. 4.148.

11) Wait until next sampling time, go to 7 and repeat.

4.6.4 Simulation and Experimental Results

The MRAC algorithm was simulated under the same cutting conditions as PID and Pole-Placement algorithms. The results for peak cutting force and control input are shown in Fig.4.35. The same algorithm in the presence of run-out was simulated and the results are shown in Fig.4.36. The algorithm which has an overall satisfactory response, shows some instability when exposed to run-out. This is significant at the very beginning of the cutting. Note that in the MRAC algorithm five parameters are estimated. The same problem has been reported by Oh [40], who has used the same method[21], although his model was first order with two parameters.

The MRAC controller was implemented on the CNC machine, using the same work-piece geometry and cutting conditions, as PID and Pole-Placement designs. The measured cutting force, control input, measured feed-rate from tacho and first estimated parameter b_0 are shown in Figs. 4.37, 4.38, 4.39 and 4.40, respectively. The controller is highly oscillatory at the beginning which affects the measured force. For the rest of the cut, although the force is tracking the reference input, the response is not ideal.

Note that the control scheme developed above is suitable for varying reference input because in a fixed input environment, such as our work with constant reference force, the model output after a few iterations reaches the final value and remains constant. This approach which is described as " pure adaptive " [22] could be altered by introducing a feed-back controller around the MRAC controller. The reference input to the model is now the error between the measured force and reference force input. For this case since the model is in a closed loop block the compensating gain for the dc offset is removed and an integral controller with the gain of 0.1 is introduced. This integrator will eliminate the steady state error. The controller is implemented on the machine and the results are shown in Figs. 4.41, 4.42 and 4.43. Although the response for the

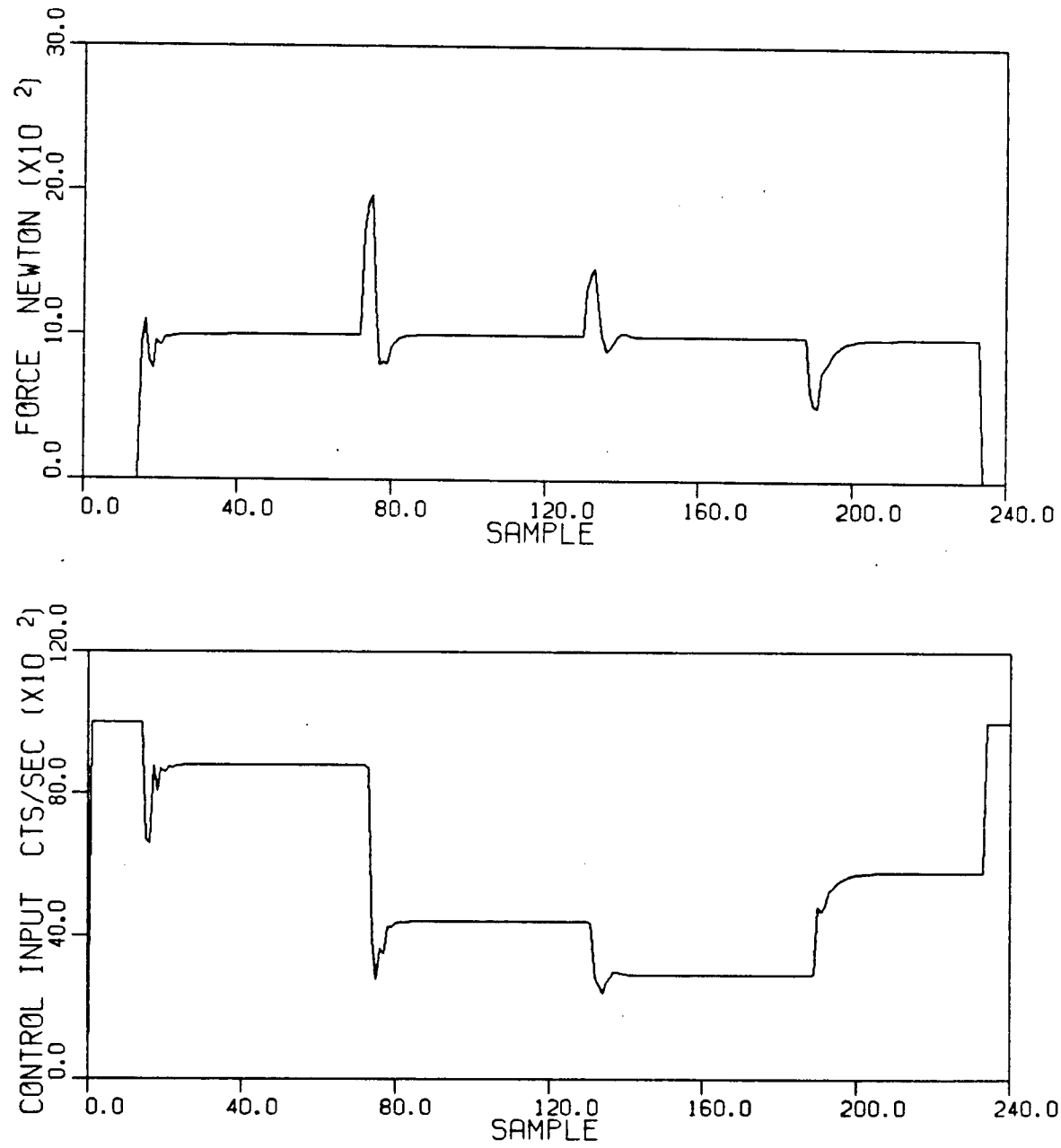

Figure 4.33: MRAC Simulation
a) Cutting Force Response
b) Control Input

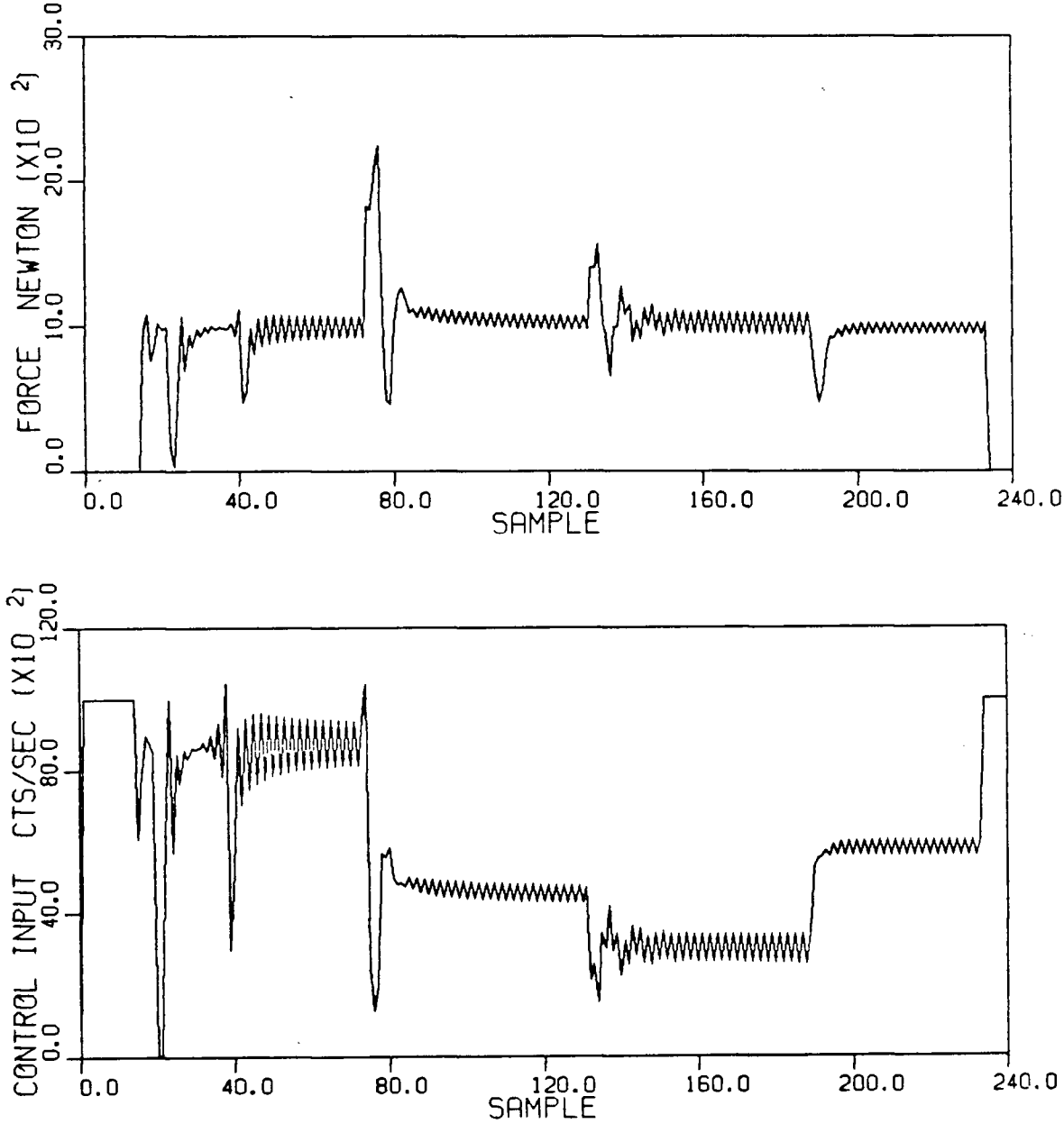

Figure 4.34: MRAC Simulation with Run-out
a) Cutting Force Response
b) Control Input

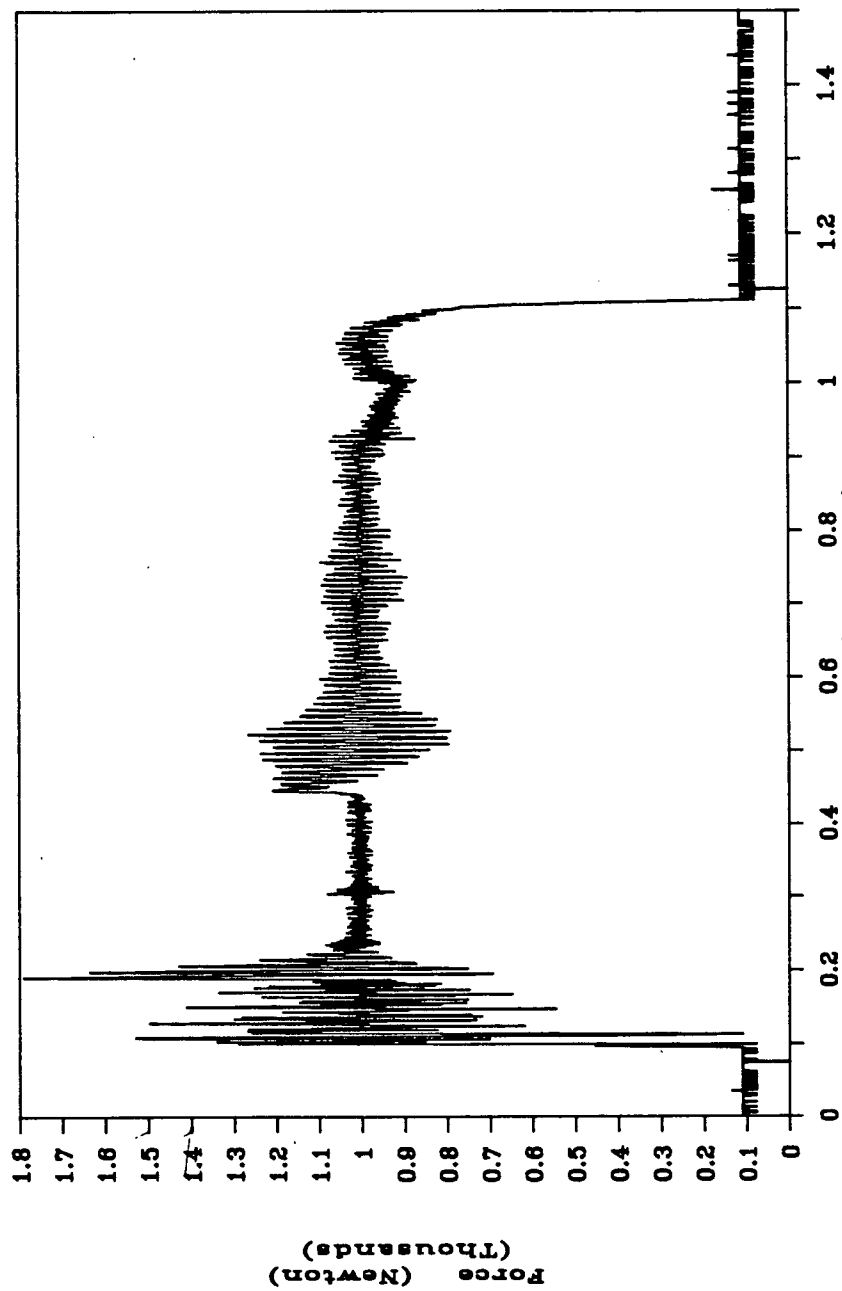

Figure 4.35: MRAC Machining Test, Measured Peak Force

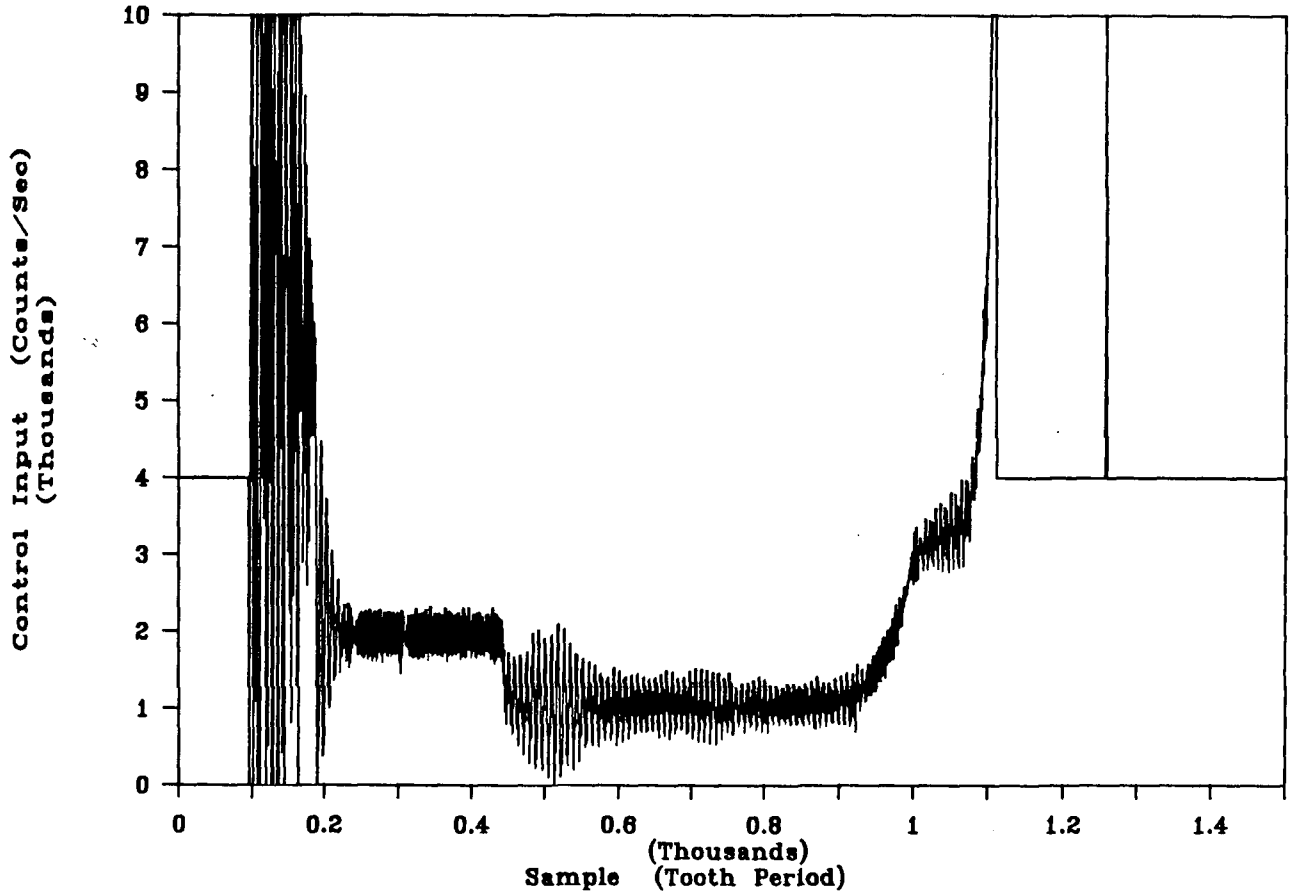

Figure 4.36: MRAC Machining Test, Control Input

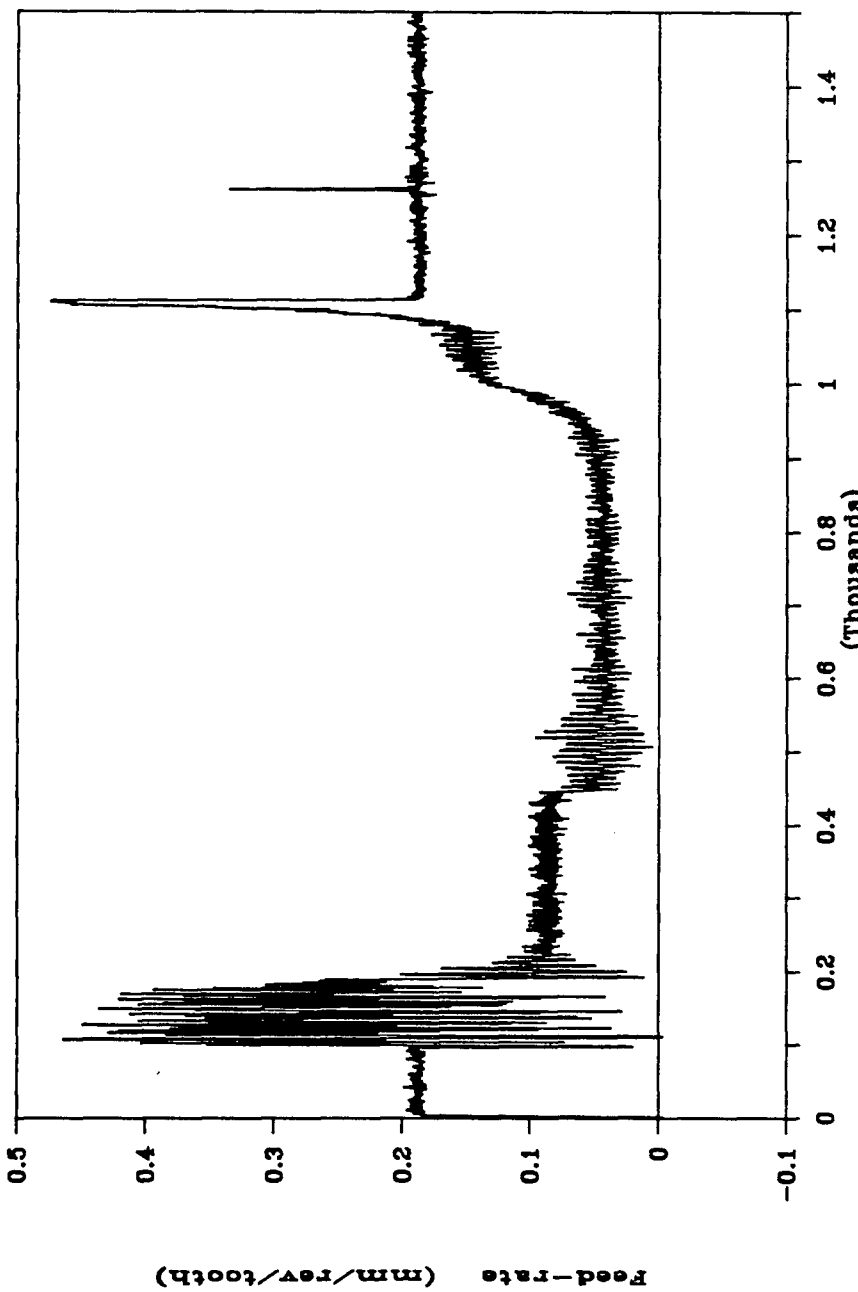

Figure 4.37: MRAC Machining Test, Measured Feed-rate

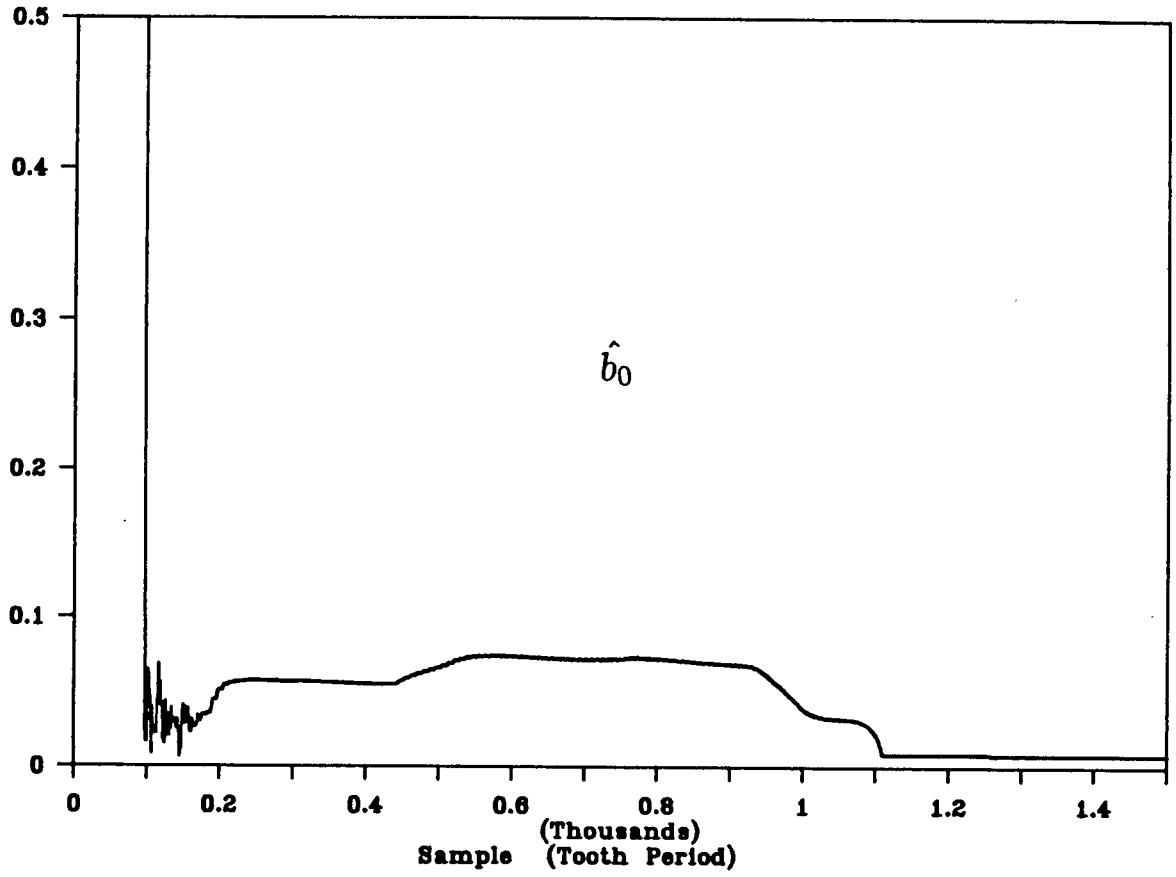

Figure 4.40: MRAC Machining Test, Estimated Parameter

start of the cutting has improved but the controller is oscillatory for the rest of the cut. Note that decreasing the integral gain does not improve the response.

The source code for the MRAC algorithm is given in Appendix C .

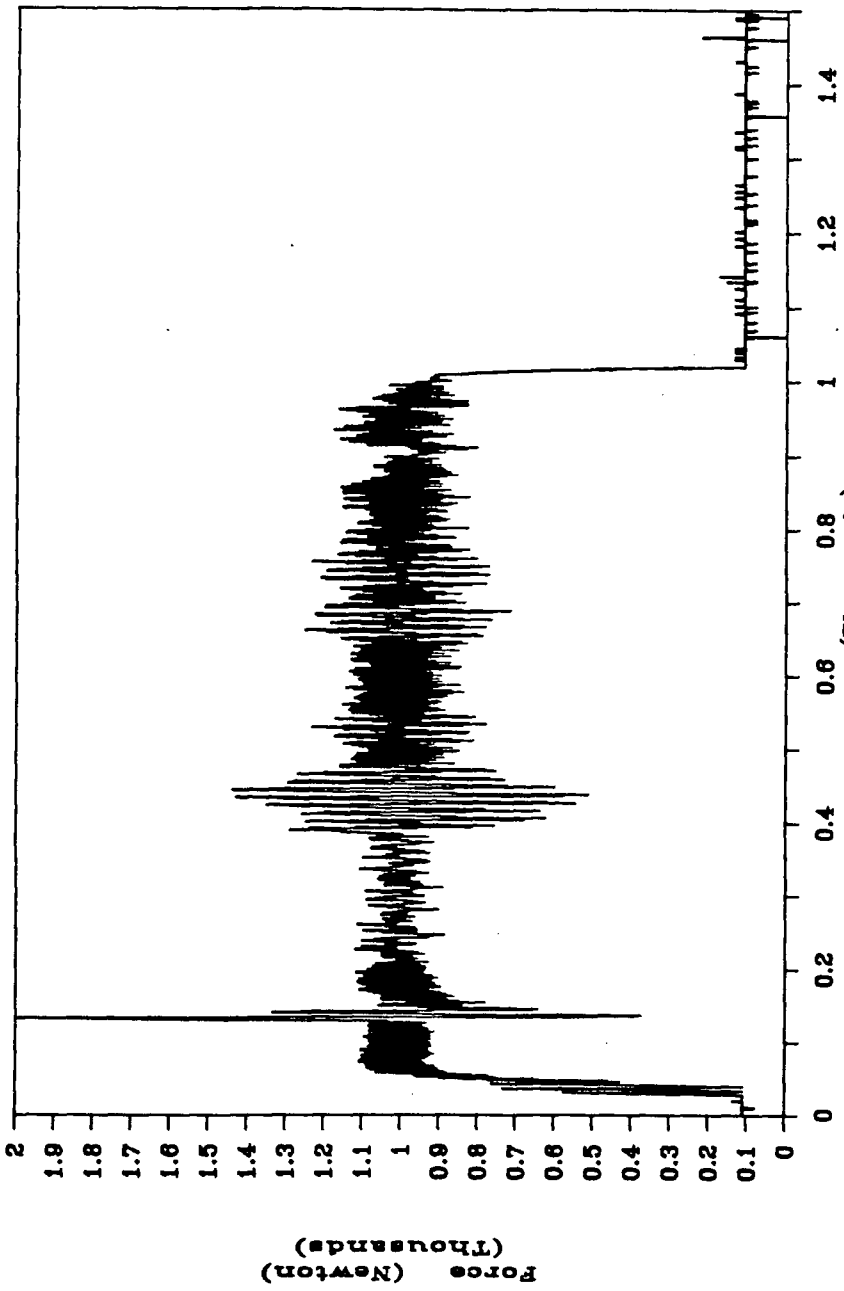

Figure 4.38: MRAC with Integral Action, Measured Peak Force

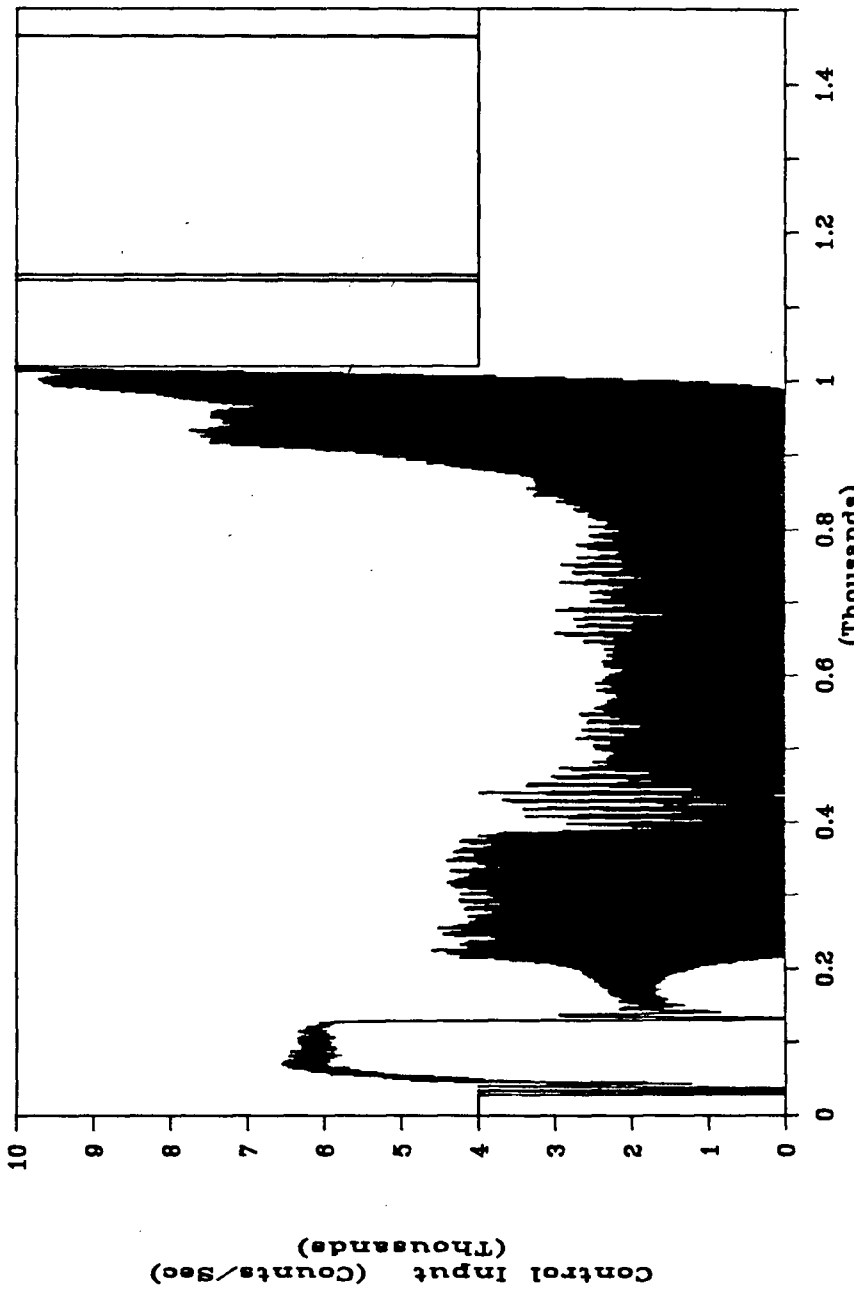

Figure 4.39: MRAC with Integral Action, Control Input

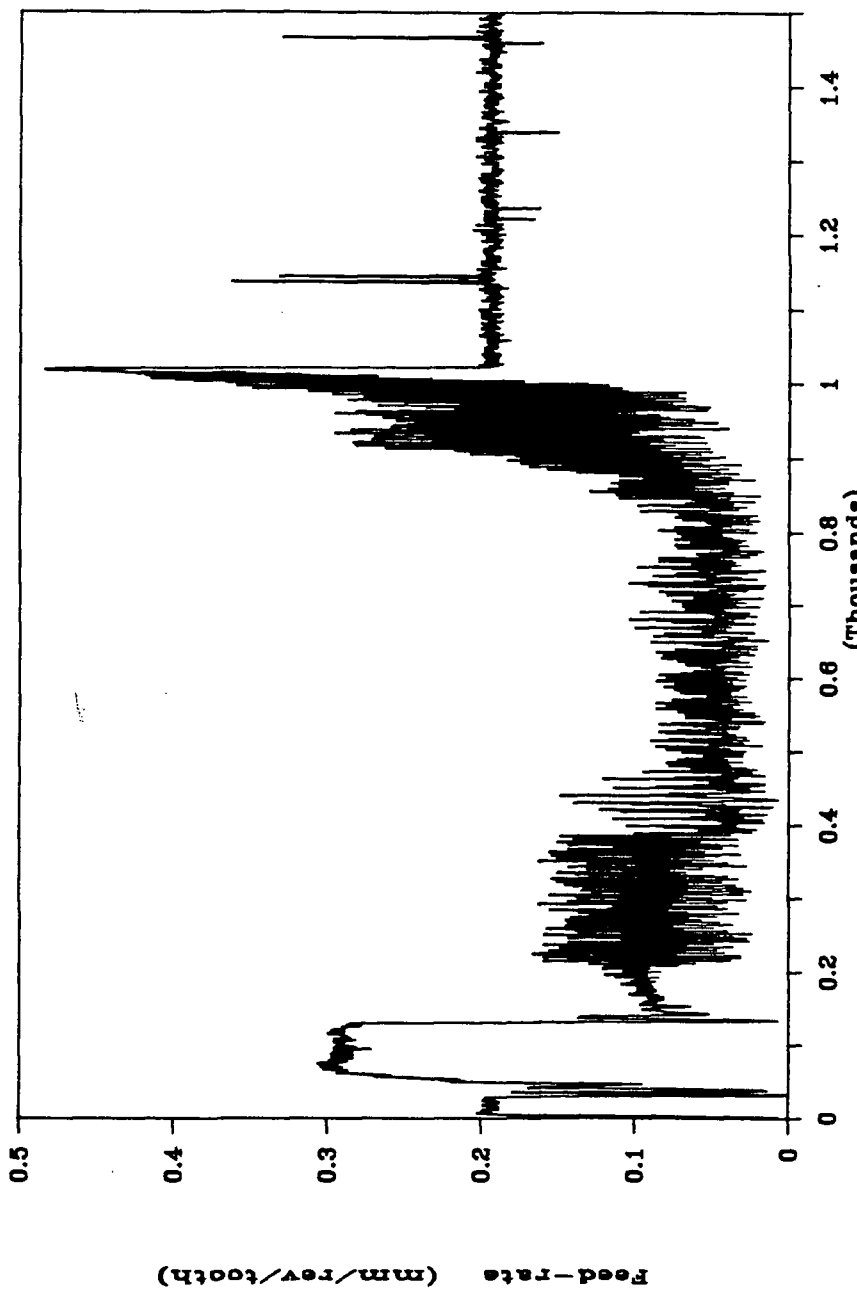

Figure 4.40: MRAC with Integral Action, Measured Feed-rate

Chapter 5

Concluding Remarks

Summary

The purpose of the work presented in this thesis was to implement adaptive control algorithms on an end milling process. The work can be summarized as following.

Modelling

A first order model was developed for the milling process. This model, only in special cases, could be reduced to a simple gain. For the feeding velocity servo a second order model was derived which was accurate enough to be used in controller design

Adaptive PID Design

The performance of the PID controller was satisfactory. One drawback of this type of controller is that a thorough knowledge of the plant is necessary in order to adjust the gains for stability.

Adaptive Pole-Placement Design

Pole-Placement controller had a satisfactory response. Selection of the poles gives an advantage to this design. The performance of both PID and Pole-Placement is subject to correct modelling of the process and correct and fast convergence of the estimator.

MRAC Design

The response of the MRAC design was oscillatory and inferior compared to self-tuning designs.

Recommendations for Future work

The work presented in this thesis requires further research in several areas. The future work is outlined as follows:

1) The modelling of the milling process should be improved. More work needs to be done to include the effect of the edge forces.

2) The effect of the unmodelled dynamics (measurement noise and run-out) should be investigated and if necessary included in the model.

3) Identification method should be improved to guarantee correct and fast convergence.

4) Work is needed to study the effect of unmodelled dynamics on MRAC design.

5) Cutting force was used as the constraint to prevent tool breakage. More work is needed to find the best constraint(s) to optimize the quality and economic factors at the same time.

6) The algorithms can not be used to prevent tool breakage due to large force overshoots during transients and should be combined with other sensing mechanisms.

7) The algorithms should be tested for more complex geometries. The work could also be extended to other types of the machine tools.

8) Force sensor which was used in this work is not practical in real situations. More research is needed to replace this by sensors on the spindle or the leadscrew, or use other signals such as current to measure the force.

Bibliography

[1] Koren,Y.,*Computer Control of Manufacturing Systems*, McGraw-Hill,New York, (1983).

[2] Tomizuka,M. and Zhang,S.," Modelling and Conventional / Adaptive PI Control of a Lathe Cutting Process," *ASME , Journal of Dynamic Systems, Measurement and Control*, Vol.110, pp.350-354 (1988).

[3] Groover,M.P., *Automation, Production Systems, and Computer Integrated Manufacturing*, Prentice-Hall,(1987).

[4] Porter,B. and Summers R.D.M.J.," Adaptive Machine - Tool Control - The State of the Art,"*Journal of Mach. Prod. Eng.* , pp.214-220 (1969).

[5] Astrom,K.J.," Theory and Applications of Adaptive Control- A Survey," *Automatica* ,Vol.19,pp. 471-486,(1983).

[6] Seborg,D.E.,Edgar,T.F. and Shah,S.L.," Adaptive Control Strategies for Process control: A Survey," *AIChE Journal*,Vol. 32, No.6, pp.881-913,(1986).

[7] Astrom,K.J.," Adaptive Feedback controls," *proceedings of the IEEE*,Vol.75, No.2, pp.185-217,(1987).

[8] Whitaker,H.P.,Yamron,J. and Kezer,A.,"Design of Model Reference Adaptive Control Systems for Aircraft," Report R-164, Instrumentation lab., MIT, Cambridge,MA(1958).

[9] Kalman,R.E.,"Design of a Self-Optimizing Control Systems,"*Transaction of ASME*,80, Ser.D, (1958).

[10] Centner,R.M. and Idelson,J.M.," Adaptive controller for a metal cutting process," *Proc. of IEEE ASME joint Automatic Control Conference*,pp.154-161,(1964).

[11] Bedini,R. and pinotti,P.C.,"A hard-wired Logic for the Adaptive Control of a Milling Machine," *International Journal of Machine Tool Design and Research*, Vol.16, pp.193-207,(1976).

[12] Koren,Y.,"Flank Wear Model of Cutting Tools Using Control Theory,"*ASME Journal of Engineering for Industry*,Vol.100, pp.103-109,(1978).

[13] Yen,D.W.and Wright,P.K.," Adaptive Control in Machining- A New Approach Based on the Physical Constraint of Tool Wear Mechanisms,"*ASME Journal of Engineering for Industry*, Vol.105, No.1,pp. 31-38,(1983).

[14] Ulsoy,A.G.,Koren,Y. and Rasmussen,F.,"Principal developments in the Adaptive Control of Machine Tools," *ASME Journal of Dynamic Systems, Measurement and Control*, Vol.105,pp.107-112(1983).

[15] Watanabe,T.,"A Model-Based Approach to Adaptive Control Optimization in Milling,"*ASME Journal of Dynamic Systems, Measurement and Control* vol. 108, pp.56-64 (1986).

[16] Beadle,B.R. and Bollinger,J.G.," Computer Adaptive Control of a Machine Tool,"*Annals of CIRP* Vol. 16, pp.61-65 (1971).

[17] Stute,G. and Goetz,F.R.,"Adaptive Control System for Variable Gain in ACC Systems,"*Proceedings of the Sixteenth International Machine Tool Design and Research Conference*,pp.117-121(1975).

[18] Tlusty,J. and Elbestawi,M.A.A.,"Analysis of Transients in an Adaptive Control Servomechanism for Milling with Constant Force,"*ASME Journal of Engineering for Industry*,Vol.99, No.3, pp.766-772 (1977).

[19] Tomizuka,M., Oh,J.H. and Dornfeld,D.A.,"Model Reference Adaptive Control of the Milling Process," in*Control of Manufacturing Processes and Robotics Systems*, Edited by D.E.Hardt and W.J.Book,ASME, New York (1983).

[20] Masory,O. and Koren,Y.," Stability Analysis of a constant Force Adaptive Control System for Turning,"*ASME Journal of Engineering for Industry*, Vol.107, pp.295-300 (1985).

[21] Landau, I.D., and Lozano, R.,"Unification of Discrete Time Model Reference Adaptive Control Designs,"*Automatica*, Vol.17, No. 4, pp.593-611 (1981).

[22] Daneshmand,L.K. and Pak,H.A.,"Model Reference Adaptive Control of Feed Force in Turning,"*ASME Journal of Dynamic Systems, Measurements, and Control*, Vol.108, pp.215-222,(1986).

[23] Lauderbaugh, L.K., and Ulsoy,A.G.,"Dynamic Modelling for Control of Milling Process,"*Proceedings of the ASME Winter Annual Meeting, Sensors and Controls for Manufacturing*, PED-Vol. 18, pp.149-158 (1985).

[24] Elbestawi,M.A. and Sagherian,R.,"Parameter Adaptive Control in Peripheral Milling,"*International Journal of Machine Tools and Manufacturing*, Vol.27, No.3, pp.399-414 (1987).

[25] Mohamed,Y.,Liu,L.and Elbestawi,M.A.,"Stochastic Self-Tuning Control of Cutting Forces in a CNC Machine Tool,"*Proceedings of IEEE Conference on Decision and Control*, pp. 1178-1183 (1988).

[26] Fussel,B.K. and Srinivasan,K.,"Model reference Adaptive Control of Force in End Milling Operations,"*Proceedings of IEEE Conference on Decision and Control*, pp. 1189-1194 (1988).

[27] Goodwin,G.C. and Sin, K.S.,*Adaptive Filtering Prediction and Control*, Prentice-Hall Inc.,New Jersey (1984).

[28] Koenigsberger,F. and Tlusty,J.,*Machine Tool Structure* Volume One,pp.77-78,Pergamon Press(1970).

[29] Altintas,Y. and Peng,J.,"Design and Analysis of a Modular CNC System for Machining Control and Monitoring,"Submitted for Publication to*ASME , Journal of Engineering for Industry*, (1988).

[30] DMC-230 Motion Control Unit Manual,*Galil Motion Control Inc.*, Palo Alto, CA (1987).

[31] Baldor DC Servo Motor Amplifier Manual,*Baldor Servo Products*, Fremont, CA (1987).

[32] Altintas,Y., and Yellowley,I.," In-Process Detection of Tool Failure in Milling Using Cutting Force Models," *ASME Journal of Engineering for Industry* to be published,(1989).

[33] Ogata,K., *Discrete-Time Control Systems*,Prentice-Hall Inc.,(1987).

[34] Altintas, Y.," Instrumentation of a Research Machine Tool with a Modular Computer Numerical Controller (CNC)," submitted to *Trans of the Institute of Measurement and Control*,(1988).

[35] Ljung,L. and Soderstrom,T.,*Theory and Practice of Recursive Identification* MIT Press,(1987).

[36] Wellsead,K.J.,Prager,D. and Zanker,P.,"Pole-Assignment Self-Tuning Regulator,"*Proc. IEE*,126(1979).

[37] Astrom,K.J. and Wittenmark,B.,*Computer Controlled Systems, Theory and Design*, Prentice-Hall,(1984).

[38] Franklin G.F. and Powel,D.,*Digital Control of Dynamic Systems*,Addison-Wesley,(1981).

[39] Lauderbaugh,L.K.," Implementation of Model Reference Adaptive Force Control in Milling," P.hD. Thesis, University of Michigan,(1985).

[40] Oh,J.," Model Reference Adaptive Control of the Milling Process," P.hD. Thesis,(1985).

[41] Astrom K.J. and Wittenmark,B., *Adaptive Control*,Addison-Wesley,(1989).

Appendix A

PID Control Algorithm

```
'%%%%%%%%%%  ADAPTIVE PID ALGORITHM %%%%%%%%%%%%
'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
DECLARE SUB SAVE.DATA (FILE.NAME$, COUNTER%)
DECLARE SUB SETUP ()
DECLARE SUB ALGORITHM ()
DECLARE SUB DATA.ACQUISITION ()
DECLARE SUB SENDBACK (CHAR%(), NUM.OF.CHAR%)
DECLARE SUB VT100 ()
DECLARE SUB INPUTINFO (GAIN%, FILE.NAME$)
DIM FI(2), P(4), TETA(2)
DIM CHAR%(50), FORCE.MAG!(1500), FEED.VEL!(1500)
DIM CONTROL.CPS!(1500), TETA.ZERO!(1500), TETA.ONE!(1500)
CLS
CALL VT100
CLS
COUNTER% = 0
CALL SETUP
TOTAL.FORCE! = 0
FEED! = 0
FI(0) = TOTAL.FORCE!
FI(1) = FEED!
P(0) = 1000!
P(3) = 1000!
P(1) = 0!
P(2) = 0!
IC% = 0
JC% = 0
HCPS! = 1000!
TETA(0) = .5
TETA(1) = .5
L.ERRER = 0
LL.ERRER = 0
LAST.CPS! = 0
OUT &H307, &H80            'TURN PC LINK RESET ON
OUT &H307, &H0             'TURN PC LINK RESET OFF
OUT &H305, &HE0            'ENABLE SYSTEM,GET BUS,SELECT I/O
START.TIME! = TIMER
DO
     COUNTER% = COUNTER% + 1
     CALL DATA.ACQUISITION
     CALL ALGORITHM
     CPS& = INT(CPS!)
     CPS$ = STR$(CPS&)
     LENGTH% = LEN(CPS$)
     CHAR%(1) = 83                        'S
     CHAR%(2) = 80                        'P
     FOR COUNT% = 1 TO LENGTH%
          CHAR$ = MID$(CPS$, COUNT%, 1)
```

```
     CHAR%(3 + LENGTH%) = 13          'CARRIAGE RETURN
     NUM.OF.CHAR% = 3 + LENGTH%
     CALL SENDBACK(CHAR%(), NUM.OF.CHAR%)
     LL.ERRER = L.ERRER
     L.ERRER = ERRER
     LAST.CPS! = CPS!
     FI(0) = TOTAL.FORCE!
     FI(1) = FEED!
     FI0 = FI(0)
     FI1 = FI(1)
LOOP UNTIL COUNTER% > 1499
FINISH.TIME! = TIMER
CHAR%(1) = 83                              'S
CHAR%(2) = 84                              'T
CHAR%(3) = 13                              '<CR>
NUM.OF.CHAR% = 3
CALL SENDBACK(CHAR%(), NUM.OF.CHAR%)
CLS
PRINT "CNC MACHINE STOPPED"
CALL VT100
PRINT (FINISH.TIME! - START.TIME!) / COUNTER%
CALL INPUTINFO(GAIN%, FILE.NAME$)
CALL SAVE.DATA(FILE.NAME$, COUNTER%)
END

SUB ALGORITHM
     SHARED CONTROL.CPS!(), TETA.ZERO!(), TETA.ONE!()
     SHARED TOTAL.FORCE!, FEED!, P(), FI(), TETA()
     SHARED COUNTER%, CPS!, LAST.CPS!, HCPS!
     SHARED ERRER, L.ERRER, LL.ERRER, IC%, JC%
     DIM S(2), K(2)
     FL = .995
     Z% = 2
     RPM% = 775
     ST = .0387
     KT = 60 / 20 * 25.4 / 1000 / RPM% / Z%
     VALU = 1000!
     REF = 1000!
     KV = .35 / ST
     IC% = IC% + 1
     JC% = JC% + 1
     ERRER = REF - TOTAL.FORCE!
     IF TOTAL.FORCE! > 150 THEN
          DEN = FI(0) * (FI(0) * P(0) + FI(1) * P(1))
          + FI(1) * (FI(0) * P(2) + FI(1) * P(3)) + FL
          S(0) = P(0) * FI(0) + P(1) * FI(1)
          S(1) = P(2) * FI(0) + P(3) * FI(1)
          K(0) = S(0) / DEN
          K(1) = S(1) / DEN
          ERRS = TOTAL.FORCE! - FI(0) * TETA(0)
          - FI(1) * TETA(1)
          P(0) = (P(0) - S(0) * S(0) / DEN) / FL
          P(2) = (P(2) - S(1) * S(0) / DEN) / FL
          P(1) = P(2)
          P(3) = (P(3) - S(1) * S(1) / DEN) / FL
          TETA(0) = TETA(0) + K(0) * ERRS
          TETA(1) = TETA(1) + K(1) * ERRS
          IF ABS(ERRS) > REF / 20! AND JC% > 25 THEN
               JC% = 0
               P(0) = VALU
               P(3) = VALU
```

```
3
                FL = .995
            ELSE
                FL = 1
            END IF
            '
            IF TETA(0) > 1 THEN
                TETA(0) = 1!
            END IF
            '
            IF TETA(1) < .001 THEN TETA(1) = .001
            SUM = KV * ST / TETA(1) / KT / .9538
            KD = SUM * TETA(0) * .0461992
            KP = (TETA(0) + .0461992) * SUM - 2! * KD
            KI = SUM - KP - KD
            CPS! = LAST.CPS! + SUM * ERRER +
            (-KP - 2! * KD) * L.ERRER + KD * LL.ERRER
        ELSE
            CPS! = 4000
        END IF
        '
        HCPS! = CPS!
        '
10      CPS! = HCPS!
        IF CPS! < 10 THEN CPS! = 10
        IF CPS! > 10000 THEN CPS! = 10000
        '
        '
CONTROL.CPS!(COUNTER%) = CPS!
TETA.ZERO!(COUNTER%) = TETA(0)
TETA.ONE!(COUNTER%) = TETA(1)
END SUB
```

Appendix B

Pole-Placement Control Algorithm

```
' %%%%%%  ADAPTIVE POLE PLACEMENT ALGORITHM  %%%%%%
         VERSION 2
'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
DECLARE SUB SAVE.DATA (FILE.NAME$, COUNTER%)
DECLARE SUB SETUP ()
DECLARE SUB ALGORITHM ()
DECLARE SUB DATA.ACQUISITION ()
DECLARE SUB SENDBACK (CHAR%(), NUM.OF.CHAR%)
DECLARE SUB VT100 ()
DECLARE SUB INPUTINFO (GAIN%, FILE.NAME$)
DIM FI(2), P(4), TETA(2)
DIM CHAR%(50), FORCE.MAG!(1500), FEED.VEL!(1500)
DIM CONTROL.CPS!(1500), TETA.ZERO!(1500), TETA.ONE!(1500)
CLS
CALL VT100
CLS
COUNTER% = 0
IC% = 0
JC% = 0
KC% = 0
CALL SETUP
TOTAL.FORCE! = 0
L.FORCE = 0
FEED! = 0
FI(0) = TOTAL.FORCE!
FI(1) = FEED!
P(0) = 1000
P(3) = 1000
P(1) = 0
P(2) = 0
TETA(0) = .5
TETA(1) = .5
L.ERRER = 0
LL.ERRER = 0
LAST.CPS! = 0
LL.CPS! = 0
HCPS! = 1000!
OUT &H307, &H80                'TURN PC LINK RESET ON
OUT &H307, &H0                 'TURN PC LINK RESET OFF
OUT &H305, &HE0                'ENABLE SYSTEM,GET BUS,SELECT I/O
START.TIME! = TIMER
DO
     COUNTER% = COUNTER% + 1
     CALL DATA.ACQUISITION
     CALL ALGORITHM
     CPS& = INT(CPS!)
     CPS$ = STR$(CPS&)
     LENGTH% = LEN(CPS$)
```

```
     FOR COUNT% = 1 TO LENGTH%
          CHAR$ = MID$(CPS$, COUNT%, 1)
          CHAR%(2 + COUNT%) = ASC(CHAR$)
     NEXT COUNT%
     CHAR%(3 + LENGTH%) = 13              'CARRIAGE RETURN
     NUM.OF.CHAR% = 3 + LENGTH%
     CALL SENDBACK(CHAR%(), NUM.OF.CHAR%)
     LL.ERRER = L.ERRER
     L.ERRER = ERRER
     LL.CPS! = LAST.CPS!
     LAST.CPS! = CPS!
     L.FORCE! = TOTAL.FORCE!
     FI(0) = TOTAL.FORCE!
     FI(1) = FEED!
LOOP UNTIL COUNTER% > 1499
FINISH.TIME! = TIMER
CHAR%(1) = 83                             'S
CHAR%(2) = 84                             'T
CHAR%(3) = 13                             '<CR>
NUM.OF.CHAR% = 3
CALL SENDBACK(CHAR%(), NUM.OF.CHAR%)
CLS
PRINT "CNC MACHINE STOPPED"
CALL VT100
PRINT (FINISH.TIME! - START.TIME!) / COUNTER%
CALL INPUTINFO(GAIN%, FILE.NAME$)
CALL SAVE.DATA(FILE.NAME$, COUNTER%)
END

SUB ALGORITHM
     SHARED CONTROL.CPS!(), TETA.ZERO!(), TETA.ONE!()
     SHARED TOTAL.FORCE!, FEED!, P(), FI(), TETA()
     SHARED COUNTER%, CPS!, LAST.CPS!, LL.CPS!, HCPS!, IC%
     SHARED ERRER, L.ERRER, LL.ERRER, L.FORCE!, KC%, JC%
     DIM S(2), K(2)
     FL = .995
     Z% = 2
     RPM% = 775
     ST = .0387
     KT = 60 / 20 * 25.4 / 1000 / RPM% / Z%
     VALU = 1000
     IC% = IC% + 1
     JC% = JC% + 1
     IF TOTAL.FORCE! > 150 THEN
       KC% = KC% + 1
       IF KC% < 25 THEN
       REF = 1000 * (1! - EXP(-COUNTER% / 5!))
       REF = 1000!
       ELSE
       REF = 1000!
       END IF
     P1 = -1.162
     P2 = .4132
     ERRER = REF - TOTAL.FORCE!
          DEN = FI(0) * (FI(0) * P(0) + FI(1) * P(1))
          + FI(1) * (FI(0) * P(2) + FI(1) * P(3)) + FL
          S(0) = P(0) * FI(0) + P(1) * FI(1)
          S(1) = P(2) * FI(0) + P(3) * FI(1)
          K(0) = S(0) / DEN
          K(1) = S(1) / DEN
          ERRS = TOTAL.FORCE! - FI(0) * TETA(0)
```

```
        IC% = 0
        GOTO 10
    END IF
    IF ABS(ERRS) > REF / 20! AND JC% > 25 THEN
        P(0) = VALU
        P(3) = VALU
        P(2) = 0
        P(1) = 0
        FL = .995
        JC% = 0
    ELSE
        FL = 1!
    END IF
        P(0) = (P(0) - S(0) * S(0) / DEN) / FL
        P(2) = (P(2) - S(1) * S(0) / DEN) / FL
        P(1) = P(2)
        P(3) = (P(3) - S(1) * S(1) / DEN) / FL
        TETA(0) = TETA(0) + K(0) * ERRS
        TETA(1) = TETA(1) + K(1) * ERRS
        '
         IF TETA(1) < 0 THEN
            TETA(1) = .5
         END IF
        '
        IF TETA(0) > 1 THEN
           TETA(0) = 1
        END IF

        K1 = TETA(1) * .9518086 * KT
        K2 = (1 + P1 + P2) / K1
        R1 = P1 + TETA(0) + .0461992
        SZ = (.0461992 * (R1 - TETA(0)) +
        TETA(0) * R1 + P2) / K1
        S1 = (-.0461992 * TETA(0) * R1) / K1
        CPS! = -(.0020946 + R1) * LAST.CPS! -
        .0020946 * R1 * LL.CPS! + K2 * REF -
        SZ * TOTAL.FORCE! - S1 * L.FORCE!
    ELSE
        CPS! = 4000
    END IF
        HCPS! = CPS!
    '
10      CPS! = HCPS!
    '
    IF CPS! < 10 THEN CPS! = 10
    IF CPS! > 10000 THEN CPS! = 10000
    '
CONTROL.CPS!(COUNTER%) = CPS!
TETA.ZERO!(COUNTER%) = TETA(0)
TETA.ONE!(COUNTER%) = TETA(1)
END SUB
```

Appendix C

MRAC Algorithm

```
' %%%%%%%  MRAC ALGORITHM %%%%%%%
'          VERSION 4  (SHORT)
' %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
DECLARE SUB SAVE.DATA (FILE.NAME$, COUNTER%)
DECLARE SUB SETUP ()
DECLARE SUB ALGORITHM ()
DECLARE SUB DATA.ACQUISITION ()
DECLARE SUB SENDBACK (CHAR%(), NUM.OF.CHAR%)
DECLARE SUB VT100 ()
DECLARE SUB INPUTINFO (GAIN%, FILE.NAME$)
DECLARE SUB DOT (XX(), YY(), SS!, NN%)
DIM P(5), PP(5), FI(5), FI1(5), FI2(5)
DIM F(5), PZ(4), FIZ(4), H(5), T(5), G(5), GG(5)
DIM CHAR%(50), FORCE.MAG!(1500), FEED.VEL!(1500)
DIM CONTROL.CPS!(1500), TETA.ZERO!(1500), TETA.ONE!(1500)
CLS
CALL VT100
CLS
COUNTER% = 0
IC% = 0
CALL SETUP
'
N% = 5
M% = N% - 1
D% = 2
VALU = 10!
TOTAL.FORCE! = 0!
LAST.FORCE! = 0!
LL.FORCE! = 0!
LAST.CPS! = 0!
LL.CPS! = 0!
FOR I = 1 TO N%
    P(I) = .5
    PP(I) = 0!
    FI(I) = 0!
    FI1(I) = 0!
    FI2(I) = 0!
    F(I) = VALU
NEXT I
'
OUT &H307, &H80                 'TURN PC LINK RESET ON
OUT &H307, &H0                  'TURN PC LINK RESET OFF
OUT &H305, &HE0                 'ENABLE SYSTEM,GET BUS,SELECT I/O
START.TIME! = TIMER
DO
     COUNTER% = COUNTER% + 1
     CALL DATA.ACQUISITION
```

```
      CPS$ = STR$(CPS&)
      LENGTH% = LEN(CPS$)
      CHAR%(1) = 83                          'S
      CHAR%(2) = 80                          'P
      CHAR%(3) = 44                          ',
      '
      FOR COUNT% = 1 TO LENGTH%
           CHAR$ = MID$(CPS$, COUNT%, 1)
           CHAR%(3 + COUNT%) = ASC(CHAR$)
      NEXT COUNT%
      CHAR%(4 + LENGTH%) = 13                'CARRIAGE RETURN
      NUM.OF.CHAR% = 4 + LENGTH%
      CALL SENDBACK(CHAR%(), NUM.OF.CHAR%)
        FOR I = 1 TO N%
        PP(I) = P(I)
        FI2(I) = FI1(I)
        NEXT I
        '
        FI(1) = CPS!
        FI(2) = LAST.CPS!
        FI(3) = LL.CPS!
        FI(4) = TOTAL.FORCE!
        FI(5) = LAST.FORCE!
        FI2(1) = LL.CPS!
        FI2(2) = LLL.CPS!
        FI2(3) = LLLL.CPS!
        FI2(4) = LL.FORCE!
        FI2(5) = LLL.FORCE!
        '
      LLL.FORCE! = LL.FORCE!
      LL.FORCE! = LAST.FORCE!
      LAST.FORCE! = TOTAL.FORCE!
      LLLL.CPS! = LLL.CPS!
      LLL.CPS! = LL.CPS!
      LL.CPS! = LAST.CPS!
      LAST.CPS! = CPS!
      YM = YM1
      YM1 = YM2
      ERK! = ERRER!
      UM1! = UM!
'
LOOP UNTIL COUNTER% > 1499
FINISH.TIME! = TIMER
CHAR%(1) = 83                                'S
CHAR%(2) = 84                                'T
CHAR%(3) = 13                                '<CR>
NUM.OF.CHAR% = 3
CALL SENDBACK(CHAR%(), NUM.OF.CHAR%)
CLS
PRINT "CNC MACHINE STOPPED"
CALL VT100
PRINT (FINISH.TIME! - START.TIME!) / COUNTER%
CALL INPUTINFO(GAIN%, FILE.NAME$)
CALL SAVE.DATA(FILE.NAME$, COUNTER%)
END

SUB ALGORITHM
      SHARED CONTROL.CPS!(), TETA.ZERO!(), TETA.ONE!()
      SHARED TOTAL.FORCE!, LAST.FORCE!, LL.FORCE!, LLL.FORCE!
      SHARED COUNTER%, CPS!, LAST.CPS!, LL.CPS!, LLL.CPS!, LLLL.CPS!
      SHARED P(), PP(), FI(), FI1(), FI2(), UM, UM1, ERRER!, ERK!
```

```
ST = .0387
VALU! = 10!
REF = 1000!
C21 = .3
C22 = .001
LAM1 = .95
LAM2 = .95
'
C11 = -.4
C12 = .04
D1 = .3
D2 = .00062838#
'
'IC% = IC% + 1
'
IF TOTAL.FORCE! > 150 THEN
ERRER! = REF - TOTAL.FORCE!
'##########   Only  For  MRAC with  integral action ##########
' UM = UM1 + (.19) * (ERRER! + ERK!)
'#############################################################
'
UM = REF
S1! = 0!
FOR I = 1 TO N%
S1! = S1! + PP(I) * FI2(I)
NEXT I
     E = TOTAL.FORCE! + C21 * LAST.FORCE! + C22 * LL.FORCE! - S1!
      '
         FOR I = 1 TO N%
             G(I) = FI2(I) * F(I)
         NEXT I
         '
 S2! = 0!
 FOR I = 1 TO N%
 S2! = S2! + G(I) * FI2(I)
 NEXT I
 '
          FOR I = 1 TO N%
               P(I) = PP(I) + G(I) / (1 + S2!) * E
          NEXT I
      '
    FOR I = 2 TO N%
    PZ(I - 1) = P(I)
    NEXT I
    '
    FIZ(1) = CPS!
    FIZ(2) = LAST.CPS!
    FIZ(3) = TOTAL.FORCE!
    FIZ(4) = LAST.FORCE!
    '
      YM2 = -C11 * YM1 - C12 * YM +
      (D1 * UM + D2 * UM1) / (D1 + D2) * (1 + C11 + C12)
      S3 = YM2 + C21 * YM1 + C22 * YM
      '
      S4! = 0!
      FOR I = 1 TO M%
      S4! = S4! + FIZ(I) * PZ(I)
      NEXT I
      '
      IF P(1) = 0 THEN P(1) = .01
      CPS! = -(S4 - S3) / P(1)
```

```
                FOR I = 1 TO N%
                F(I) = (F(I) - G(I) * G(I) / S6) / LAM1
                NEXT I
    ELSE
        CPS! = 4000
    END IF
    '
    IF CPS! < 10 THEN CPS! = 10
    IF CPS! > 12000 THEN CPS! = 12000
    '
CONTROL.CPS!(COUNTER%) = CPS!
TETA.ZERO!(COUNTER%) = P(1)
TETA.ONE!(COUNTER%) = P(2)
END SUB
```

Appendix D

Data Acquisition Software

```
' %%%%%%%  DATA ACQUISITION BOARD SOFTWARES  %%%%%%%%%
' %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
SUB DATA.ACQUISITION
DIM VOLTAGE!(3)
     SHARED TOTAL.FORCE!, FI0, FI1, FEED!, COUNTER%
     SHARED FORCE.MAG!(), FEED.VEL!()
     ' ------------------------------------------------------------
     '                              LOOP
     ' ------------------------------------------------------------
     '
     ' This subprogram acquires feedrate and force feedback from
     ' the CNC machine
     ' Data is sent via DMA to memory to be accessed by the program
     ' when required.
     '
     ' ------------------------------------------------------------
     '
     ' Define all the constants used in the subprogram
     '
     DEFINT A-Z              ' Define all variables to be integers
     RPM% = 775
     Z% = 2
     TOO.MUCH = 100          ' Number of times the loop will be executed
     BASE.ADDRESS = &H2EC    ' Base address of DT2801-A
     COMMAND.REGISTER = BASE.ADDRESS + 1 ' Address of DT2801-A command register
     STATUS.REGISTER = BASE.ADDRESS + 1  ' Address of DT2801-A status register
     DATA.REGISTER = BASE.ADDRESS        ' Address of DT2801-A data register
     COMMAND.WAIT = &H4          ' Wait for command in ready mask
     WRITE.WAIT = &H2            ' Wait for write data in mask
     READ.WAIT = &H5             ' Wait for read data out mask
     CSTOP = &HF                 ' Stop DT2801-A board command
     CCLEAR = &H1                ' Clear DT2801-A board command
     CERROR = &H2                ' Get error from DT2801-A board command
     CCLOCK = &H3                ' Set DT2801-A board clock command
     CCONTINUOUS = 32            ' Continuous conversions command
     CSAD = &HD                  ' Set A/D parameters command
     CRAD = &HE                  ' Read A/D command
     EXT.CLOCK = &H40            ' Use external clock command
     EXT.TRIGGER = &H80          ' Use external trigger command
     CDMA = &H10                 ' Start D/A conversions command
     DUMMY = 5                   ' Dummy value
     BASE.FREQUENCY# = 800000!   ' Base frequency of the clock on the DT2801-A
     PERIOD# = 40                ' This value indicates a quick scan. See manual.
     GAIN = 1
     SEGMENT& = 32768
     ''
     '' Define DMA constants for DMA channel 1.
     ''
```

```
     ''
     '' The memory to be used for DMA starts at memory address &HF000
     '' on memory page 8.
     ''
     DMACHANNEL = 1          ' The channel to use for DMA
     DMAMODE = &H45          ' Use DMA from a peripheral to memory
     BASEREG = 2             ' Port address to send base info
     COUNTREG = 3            ' Port address to send count info
     PAGEREG = &H83          ' Port address to send page info
     DMABASEL = 0            ' Low byte of base address
     DMABASEH = 0            ' High byte of base address
     DMAPAGE = 8             ' Page to send data to

     ' ================================================
     '            SETUP DT2801-A BOARD FOR READ ON TRIGGER
     ' ================================================
     '
5103 '' Set-up DMA chip to transfer the required number of bytes.
5106 '' (1 sample x 2 channels x 2 bytes/sample ) - 1 byte
5107 ''
5109    DMACOUNT = ((1 * 2) * 2) - 1
5110    DMACOUNTH = INT(DMACOUNT / 256)
5113    DMACOUNTL = DMACOUNT - DMACOUNTH * 256
5116 ''
5119    OUT 11, DMAMODE             ' set DMA mode
5120    OUT 12, 0                   ' clear byte flip-flop
5123    OUT BASEREG, DMABASEL       ' set DMA memory base address
5126    OUT BASEREG, DMABASEH       '
5129    OUT COUNTREG, DMACOUNTL     ' set DMA byte count
5130    OUT COUNTREG, DMACOUNTH     '
5133    OUT PAGEREG, DMAPAGE        ' set DMA memory page
5136    OUT 10, DMACHANNEL          ' enable DMA channel mask
5200 ''
5210 '' Check for ERROR from board before continuing. (not essential)
5220 ''
        WAIT STATUS.REGISTER, WRITE.WAIT, WRITE.WAIT
5230    WAIT STATUS.REGISTER, COMMAND.WAIT
5240    STATUS = INP(STATUS.REGISTER)
5250    IF (STATUS AND &H80) THEN PRINT "D.A.1-ERROR"
5300 ''
6000 ''
6010 '' Write READ A/D WITH DMA command.
6015 ''
        WAIT STATUS.REGISTER, WRITE.WAIT, WRITE.WAIT
6110    WAIT STATUS.REGISTER, COMMAND.WAIT
6120    OUT COMMAND.REGISTER, CRAD + CDMA + EXT.TRIGGER
6300 ''
6310 '' Check for ERROR from board after conversions completed.
6320 '' The program will pause here until the trigger has occured
     '' and the data has all been transferred. Ie: the board is
     '' not interested in getting another command from you until
     '' it has finished the earlier one.
     ''

        WAIT STATUS.REGISTER, WRITE.WAIT, WRITE.WAIT
6330    WAIT STATUS.REGISTER, COMMAND.WAIT
6340    STATUS = INP(STATUS.REGISTER)
6350    IF (STATUS AND &H80) THEN PRINT "D.A.2-ERROR"
6400 ''
     DEF SEG = SEGMENT&
     FOR I = 1 TO 2
```

```
        DATA.VALUE = DATA.VALUE + PEEK(ADDRESS + 1) * 256
        VOLTAGE!(I) = (DATA.VALUE - 2048) * (10 / GAIN) / 2048
    NEXT I
    TOTAL.FORCE! = SQR(-VOLTAGE!(1) * 10 / 2) * 500       '500 N/mv
    FEED! = VOLTAGE!(2) * 531.32 / (RPM% * Z%)       '[mm/rev*tooth]
    DEF SEG
    FORCE.MAG!(COUNTER%) = TOTAL.FORCE!
    FEED.VEL!(COUNTER%) = FEED!
END SUB

DEFSNG A-Z
SUB INPUTINFO (GAIN%, FILE.NAME$)
    '
    ' Input the gain
    '
    DO
        LOCATE 5, 5                         ' Locate cursor at input
        INPUT "Gain : ", RESPONSE$          ' Read in the input value
        GAIN% = VAL(RESPONSE$)              ' Convert input
        IF GAIN% = 1 OR GAIN% = 2 OR GAIN% = 4 OR GAIN% = 8 THEN EXIT DO
        LOCATE 5, 5                         ' Locate cursor at input
        PRINT STRING$(60, 32)               ' Erase input
        LOCATE 23, 5                        ' Locate cursor in message window
        COLOR 0, 7
        PRINT "Gain can only be 1,2,4 or 8"  ' Print error message
        COLOR 7, 0
    LOOP
    LOCATE 23, 5                        ' Locate cursor in message window
    PRINT STRING$(60, 32)               ' Erase any message
    '
    ' Input the file name
    '
    DO
        LOCATE 9, 5                     ' Locate cursor at input
        INPUT "File Name : ", RESPONSE$ ' Read in input data
        HEAD$ = RESPONSE$               ' Convert input data
        IF LEN(HEAD$) < 6 AND LEN(HEAD$) > 0 THEN EXIT DO ' If within range then exit
        LOCATE 23, 5                    ' Locate cursor in message window
        COLOR 0, 7
        PRINT "File name has to be from 1 to 5 characters long"
        COLOR 7, 0
        LOCATE 9, 5                     ' Locate cursor at input
        PRINT STRING$(60, 32)           ' Erase input
    LOOP
    LOCATE 23, 5                        ' Locate cursor in message window
    PRINT STRING$(60, 32)               ' Erase any message
    '
    ' Input the file number
    '
    '
    DO
        LOCATE 11, 5                        ' Locate cursor at input
        INPUT "File Number : ", RESPONSE$   ' Read in value
        NUMBER% = VAL(RESPONSE$)            ' Convert value
        IF NUMBER% > 0 AND NUMBER% < 100 THEN EXIT DO ' If value within range then exit
        LOCATE 23, 5                        ' Locate cursor in message window
        COLOR 0, 7
        PRINT "Number has to be between 1 and 99"    ' Print error message
        COLOR 7, 0
        LOCATE 11, 5                        ' Locate cursor at input
        PRINT STRING$(60, 32)               ' Erase input
```

```
     IF NUMBER% < 10 THEN NUMBER$ = RIGHT$(SEMI$, 1)
     ELSE NUMBER$ = RIGHT$(SEMI$, 2)
     LOCATE 23, 5                          ' Locate cursor in message window
     PRINT STRING$(60, 32)                 ' Erase message
     '
     ' Input the destination drive
     '
     '
     LOCATE 13, 5                          ' Locate cursor at input
     INPUT "Destination Drive and Directory : ", RESPONSE$     ' Read input
     DRIVE$ = RESPONSE$                    ' Convert input

     FILE.NAME$ = DRIVE$ + HEAD$ + NUMBER$
END SUB

SUB SAVE.DATA (FILE.NAME$, COUNTER%)
     SHARED FORCE.MAG!(), FEED.VEL!(), CONTROL.CPS!()
     SHARED TETA.ZERO!(), TETA.ONE!()
     FILE1$ = FILE.NAME$ + "A.DAT"
     FILE2$ = FILE.NAME$ + "B.DAT" -
     OPEN FILE2$ FOR OUTPUT AS #1
     FOR I% = 1 TO COUNTER%
          WRITE #1, FORCE.MAG!(I%), FEED.VEL!(I%),
          CONTROL.CPS!(I%), TETA.ZERO!(I%), TETA.ONE!(I%)
     NEXT I%
     CLOSE #1

END SUB

SUB SENDBACK (CHAR%(), NUM.OF.CHAR%)
     COUNT% = 1
     DO UNTIL COUNT% > NUM.OF.CHAR%
          ' WAIT UNTIL BUFFER CLEARED
          DO
               OUT &H302, &H3
               STATUS.BYTE% = INP(&H301)
               BIT3% = INT(STATUS.BYTE% / 8)                'ERROR=1
               BIT2% = INT((STATUS.BYTE% - BIT3% * 8) / 4) 'FULL=0
               BIT1% = INT((STATUS.BYTE% - BIT3% * 8 - BIT2% * 4) / 2)'WRITE=0
          LOOP UNTIL BIT2% <> 0
          IF BIT3% = 1 THEN PRINT "S.B.-ERROR"
          IF BIT1% <> 0 THEN
               OUT &H302, &H2
               OUT &H301, CHAR%(COUNT%)
               'CHAR$ = CHR$(CHAR%(COUNT%))
               'PRINT CHAR$;
               COUNT% = COUNT% + 1
          ELSE
               DO
                    OUT &H302, &H3
                    STATUS.BYTE% = INP(&H301)
                    FINISH% = INT(STATUS.BYTE% / 2 -
                    INT(STATUS.BYTE% / 2) + .5)
                    IF FINISH% = 1 THEN EXIT DO
                    OUT &H302, &H2
                    CHAR.BACK% = INP(&H301)
               LOOP
          END IF
     LOOP

END SUB
```

```
    ' Define all the constants used in the subprogram
    '
    DEFINT A-Z                  ' Define all variables to be integers
    TOO.MUCH = 100              ' Number of times the loop will be executed
    BASE.ADDRESS = &H2EC        ' Base address of DT2801-A
    COMMAND.REGISTER = BASE.ADDRESS + 1 ' Address of DT2801-A command register
    STATUS.REGISTER = BASE.ADDRESS + 1  ' Address of DT2801-A status register
    DATA.REGISTER = BASE.ADDRESS        ' Address of DT2801-A data register
    COMMAND.WAIT = &H4          ' Wait for command in ready mask
    WRITE.WAIT = &H2            ' Wait for write data in mask
    READ.WAIT = &H5             ' Wait for read data out mask
    CSTOP = &HF                 ' Stop DT2801-A board command
    CCLEAR = &H1                ' Clear DT2801-A board command
    CERROR = &H2                ' Get error from DT2801-A board command
    CCLOCK = &H3                ' Set DT2801-A board clock command
    CCONTINUOUS = 32            ' Continuous conversions command
    CSAD = &HD                  ' Set A/D parameters command
    CRAD = &HE                  ' Read A/D command
    EXT.CLOCK = &H40            ' Use external clock command
    EXT.TRIGGER = &H80          ' Use external trigger command
    CDMA = &H10                 ' Start D/A conversions command
    DUMMY = 5                   ' Dummy value
    BASE.FREQUENCY# = 800000!   ' Base frequency of the clock on the DT2801-A
    PERIOD# = 40                ' This value indicates a quick scan. See manual.
    ''
    '' Define DMA constants for DMA channel 1.
    ''
    '' The DMAMODE value is set to program DMA channel 1
    '' for single byte transfer memory write.
    ''
    '' The memory to be used for DMA starts at memory address &HF000
    '' on memory page 8.
    ''
    DMACHANNEL = 1                  ' The channel to use for DMA
    DMAMODE = &H45                  ' Use DMA from a peripheral to memory
    BASEREG = 2                     ' Port address to send base info
    COUNTREG = 3                    ' Port address to send count info
    PAGEREG = &H83                  ' Port address to send page info
    DMABASEL = 0                    ' Low byte of base address
    DMABASEH = 0                    ' High byte of base address
    DMAPAGE = 8                     ' Page to send data to
    ''
    '' Check for legal Status Register. If not then error! (ie: no board)
    ''
    ' ------------------------------------------------------
    '                      SET UP THE BOARD
    ' ------------------------------------------------------
    ' This routine sets up the basic parameters of the DT2801-A board
    ' prior to actually doing any A/D conversions.
    ' ------------------------------------------------------
   ' CALL RESET.DT(I%)
PRINT "reset completed"
    '
    STATUS = INP(STATUS.REGISTER)
    IF NOT ((STATUS AND &H70) = 0) THEN PRINT "S.E.2-ERROR": END
    ''

    ''
    '' Stop and clear the DT2801.
    ''
```

```
     WAIT STATUS.REGISTER, WRITE.WAIT, WRITE.WAIT
     WAIT STATUS.REGISTER, COMMAND.WAIT
     OUT COMMAND.REGISTER, CCLEAR
PRINT "stopped and cleared"
     ''
     '' Set internal clock rate.
     '' Write SET CLOCK PERIOD command.
     ''
     WAIT STATUS.REGISTER, WRITE.WAIT, WRITE.WAIT
     WAIT STATUS.REGISTER, COMMAND.WAIT
     OUT COMMAND.REGISTER, CCLOCK
PRINT "clock period set"
     ''
     '' Write high and low bytes of PERIOD#.
     ''
     PERIODH = INT(PERIOD# / 256)
     PERIODL = PERIOD# - PERIODH * 256
     WAIT STATUS.REGISTER, WRITE.WAIT, WRITE.WAIT
     OUT DATA.REGISTER, PERIODL
     WAIT STATUS.REGISTER, WRITE.WAIT, WRITE.WAIT
     OUT DATA.REGISTER, PERIODH
     IF NOT ((STATUS AND &H70) = 0) THEN PRINT "S.E.2-ERROR": END
PRINT "period# written"
     ''
     '' Set the gain code for a gain of 1.
     ''
     GAIN.CODE = 0
     ''
     '' Set A/D channels.
     ''
     START.CHANNEL = 0
     END.CHANNEL = 1
     ''
     '' Set up the A/D converter.
     '' Write SET A/D PARAMETERS command.
     ''
     WAIT STATUS.REGISTER, WRITE.WAIT, WRITE.WAIT
     WAIT STATUS.REGISTER, COMMAND.WAIT
     OUT COMMAND.REGISTER, CSAD
     ''
     '' Write A/D gain byte.
     ''
     WAIT STATUS.REGISTER, WRITE.WAIT, WRITE.WAIT
     OUT DATA.REGISTER, GAIN.CODE
PRINT "gain code written"
     ''
     '' Write A/D start channel byte.
     ''
     WAIT STATUS.REGISTER, WRITE.WAIT, WRITE.WAIT
     OUT DATA.REGISTER, START.CHANNEL
     ''
     '' Write A/D end channel byte.
     ''
     WAIT STATUS.REGISTER, WRITE.WAIT, WRITE.WAIT
      OUT DATA.REGISTER, END.CHANNEL
PRINT "channels written"
     ''
     '' Write two bytes, dummy number of conversions.
     ''
     WAIT STATUS.REGISTER, WRITE.WAIT, WRITE.WAIT
     OUT DATA.REGISTER, DUMMY
```

```
PRINT "end of SETUP"
END SUB

DEFSNG A-Z
SUB VT100
     OUT &H307, &H80              'TURN PC LINK RESET ON
     OUT &H307, &H0               'TURN PC LINK RESET OFF
     OUT &H305, &HE0              'ENABLE SYSTEM,GET BUS,SELECT I/O
     LOCATE , , 1                 'TURN CURSOR ON
     PRINT ":";
     DO
          DO
               CHAR$ = INKEY$
          LOOP UNTIL CHAR$ <> ""
          CHAR% = ASC(CHAR$)
          ' WAIT UNTIL BUFFER CLEARED
          DO
               OUT &H302, &H3
               STATUS.BYTE% = INP(&H301)
               BIT3% = INT(STATUS.BYTE% / 8)
               BIT2% = INT((STATUS.BYTE% - BIT3% * 8) / 4)
               BIT1% = INT((STATUS.BYTE% - BIT3% * 8 - BIT2% * 4) / 2)
          LOOP UNTIL BIT2% <> 0
          IF BIT3% = 1 THEN PRINT "ERROR"
          IF BIT1% <> 0 THEN
               OUT &H302, &H2
               OUT &H301, CHAR%
               IF CHAR% = 8 THEN
                    LOCATE , POS(0) - 1, 0
                    PRINT " ";
                    LOCATE , POS(0) - 1, 0
               ELSE
                    PRINT CHAR$;
               END IF
          ELSE
               DO
                    OUT &H302, &H3
                    STATUS.BYTE% = INP(&H301)
                    FINI% = INT(STATUS.BYTE% / 2 - INT(STATUS.BYTE% / 2) + .5)
                    IF FINI% = 1 THEN EXIT DO
                    OUT &H302, &H2
                    CHAR.BACK% = INP(&H301)
                    CHAR.BACK$ = CHR$(CHAR.BACK%)
                    PRINT CHAR.BACK$;
               LOOP
          END IF
          IF CHAR% = 13 THEN
               DO
                    OUT &H302, &H3
                    STATUS.BYTE% = INP(&H301)
                    NOW% = INT(STATUS.BYTE% / 2 - INT(STATUS.BYTE% / 2) + .5)
               LOOP WHILE NOW% = 1
               DO
                    OUT &H302, &H2
                    CHAR.BACK% = INP(&H301)
                    CHAR.BACK$ = CHR$(CHAR.BACK%)
                    PRINT CHAR.BACK$;
                    OUT &H302, &H3
                    STATUS.BYTE% = INP(&H301)
                    FINISH% = INT(STATUS.BYTE% / 2 - INT(STATUS.BYTE% / 2) + .5)
               LOOP WHILE FINISH% = 0
                    CLS
               END SUB
```